

# 医療とAIビッグデータ応用

## NLP②

本スライドは、自由にお使いください。  
使用した場合は、このQRコードからアンケート  
に回答をお願いします。



統合教育機構  
須藤毅顕

# Huggingface Transformers

Huggingface Transformersは、世界中で使われている自然言語処理の深層学習フレームワーク。最先端の自然言語処理アルゴリズムを簡単に試すことが出来る。

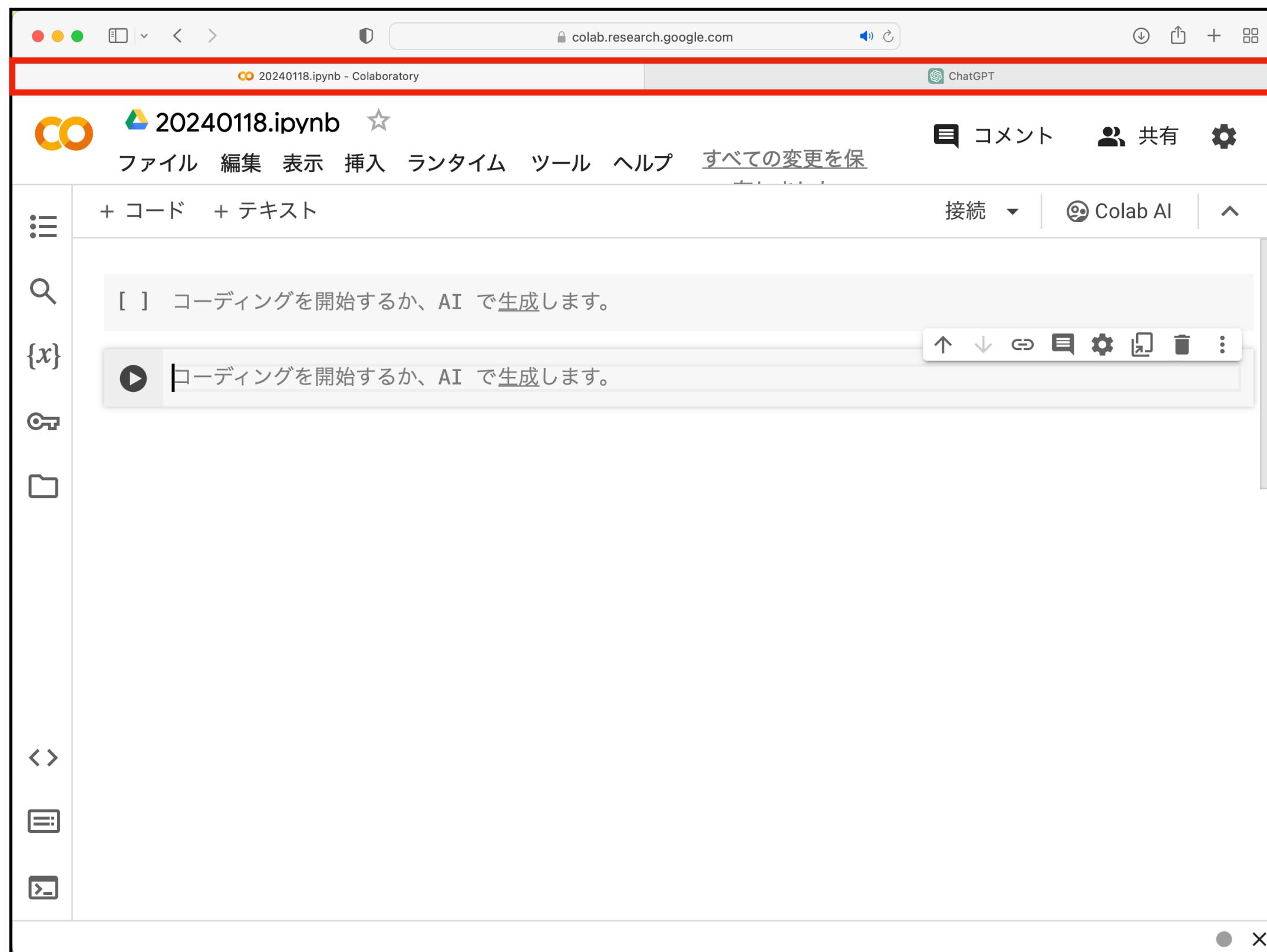
The screenshot shows the Hugging Face website homepage. At the top left is the "Hugging Face" logo with a yellow smiley face icon. Next to it is a search bar with the placeholder "Search models, datasets, users...". The top navigation bar includes links for "Models", "Datasets", "Spaces", "Docs", "Solutions", "Pricing", "Log In", and "Sign Up". Below the navigation is a large dark sidebar containing the text "The AI community building the future." in white, accompanied by a yellow smiley face emoji. To the right of the sidebar is a list of "Models" with 469,541 entries. The list includes various model names such as "meta-llama/Llama-2-70b", "stabilityai/stable-diffusion-xl-base-0.9", "openchat/openchat", "llyasviel/ControlNet-v1-1", "cerspense/zeroscope\_v2\_XL", "meta-llama/Llama-2-13b", "tiiuae/falcon-40b-instruct", "WizardLM/WizardCoder-15B-V1.0", "CompVis/stable-diffusion-v1-4", "stabilityai/stable-diffusion-2-1", and "Salesforce/xgen-7b-8k-inst". Each model entry shows its name, task (e.g., Text Generation), last update time, number of downloads, and likes.

<https://huggingface.co>

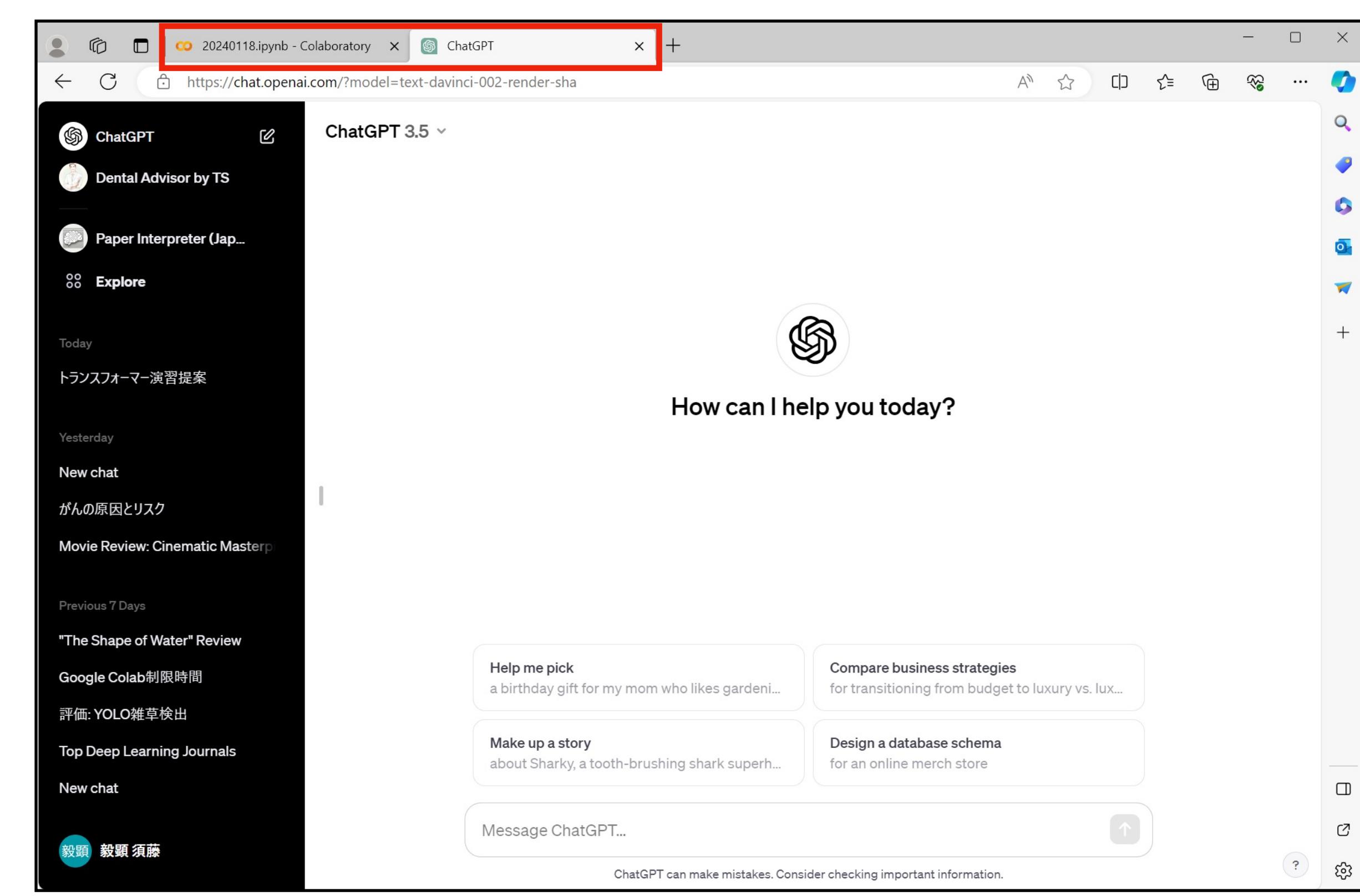
# 前準備

## Google ColabとChatGPTを立ち上げましょう

mac



Windows



タブで切り替えられるようにしましょう

# Huggingface Transformersのインストール

pip install transformers

```
pip install transformers

Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-packages (4.35.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from transformers) (3.13.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.16.4 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.19.4)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (1.23.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (23.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (6.0.1)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (2023.6.3)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformers) (2.31.0)
Requirement already satisfied: tokenizers<0.19,>=0.14 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.15.0)
Requirement already satisfied: safetensors>=0.3.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.4.1)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers) (4.66.1)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.16.4->transformers) (2023.6.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.16.4->transformers) (4.5.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2023.11.17)
```

pipは外からライブラリをインストールする命令です

これまでのkerasやmatplotlibなどはcolabに最初からインストール済み  
transformersはインストールされていないのでインストールしてからライブラリを読み込む  
(notebookを起動したら毎回インストールする必要あり)

# Huggingface Transformersの推論方法

Huggingface Transformersは、「パイプライン(pipeline)」と  
「トークナイザー+モデル」の2つ方法で推論(予測)できる

パイプライン `pipeline()` の1行のみで推論を行う最もシンプルな方法

```
(結果) = pipeline( (タスク), (モデル) )
```

トークナイザー+モデル トークナイザー：テキストをモデルの入力データに変換  
モデル：カスタムして推論

```
(モデルへの入力データ) = tokenizer.encode( (入力テキスト) )  
(結果) = model(モデルへの入力データ)
```

# Huggingface Transformersの推論方法

Huggingface Transformersは、「パイプライン(pipeline)」と  
「トークナイザー+モデル」の2つ方法で推論(予測)できる

パイプライン **pipeline()** の1行のみで推論を行う最もシンプルな方法

**(結果) = pipeline( (タスク), (モデル) )**

トークナイザー+モデル トークナイザー：テキストをモデルの入力データに変換  
モデル：カスタムして推論

**(モデルへの入力データ) = tokenizer.encode( (入力テキスト) )**  
**(結果) = model(モデルへの入力データ)**

# Pipeline()で指定できるタスク

- **text-classification** : テキスト分類
- **question-answering** : 質問応答
- **summarization** : 要約
- **text-generation** : テキスト生成
- **fill-mask** : 言語モデル(MLM)
- **ner** : 固有表現抽出
- **translation\_xx\_to\_yy**
- **text2text-generation** : text-to-text(翻訳、要約、質問応答)
- **feature-extraction** : 特徴抽出
- **zero-shot-classification** : Zero-Shot分類(未学習カテゴリでの分類)
- **convensational** : 対話
- など

# Pipeline()で指定できるタスク

- **text-classification** : テキスト分類
- **question-answering** : 質問応答
- **summarization** : 要約
- **text-generation** : テキスト生成
- **fill-mask** : 言語モデル(MLM)
- **ner** : 固有表現抽出
- **translation\_xx\_to\_yy**
- **text2text-generation** : text-to-text(翻訳、要約、質問応答)
- **feature-extraction** : 特徴抽出
- **zero-shot-classification** : Zero-Shot分類(未学習カテゴリでの分類)
- **convensational** : 対話
- など

# 自然言語処理タスク

- ・テキスト分類：「テキスト」を事前準備された「カテゴリ」に分類する

カテゴリ例：ポジティブ or ネガティブ

「昨日の映画は面白い」 → 「ポジティブ」  
「演技が下手で見ていられなかった」 → 「ネガティブ」

カテゴリ例：疾患の疑い or 健康

「彼は胸を押さえて苦しそうだ」 → 「疾患の疑い」  
「私は毎日不自由なく過ごしていく」 → 「健康」

テキスト分類：「テキスト」を事前準備された「カテゴリ」に分類する

テキスト分類のタスク名：'text-classification' or 'sentiment-analysis'

```
from transformers import pipeline  
  
nlp = pipeline('sentiment-analysis')  
text = "The movie was interesting."  
nlp(text)
```

パイプラインの読み込み

変数 = pipeline(タスク、モデル)

モデルは省略可

ラベルとスコアが得られる

[{'label': 'POSITIVE', 'score': 0.9998656511306763}]

このパイプラインのモデルは「SST-2」というデータセットで学習している  
SST-2：映画レビューがポジティブかネガティブかを分類するためのデータセット

テキスト分類：「テキスト」を事前準備された「カテゴリ」に分類する

ChatGPTに文章を作らせて、transformerでテキスト分類させよう

ChatGPT 3.5 ▾

あなた  
映画のレビューを英語で1文で書いて

ChatGPT  
"An exhilarating cinematic masterpiece that seamlessly blends captivating storytelling with breathtaking visuals."

□ ▲ □ □ □

意味が分からない？

# テキスト分類：「テキスト」を事前準備された「カテゴリ」に分類する

ChatGPTに文章を作らせて、transformerでテキスト分類させよう

ChatGPT 3.5 ▾

あなた 映画のレビューを英語で1文で書いて

ChatGPT "An exhilarating cinematic masterpiece that seamlessly blends captivating storytelling with breathtaking visuals."

□ ▲ □ □ □

ChatGPT 3.5 ▾

あなた 映画のレビューを英語で1文で書いて

ChatGPT "An exhilarating cinematic masterpiece that seamlessly blends captivating storytelling with breathtaking visuals."

あなた 日本語にして

ChatGPT "見応えある映画で、魅力的なストーリーテリングと息をのむような映像が見事に融合しています。"

意味が分からない？

ポジティブそう

# テキスト分類：「テキスト」を事前準備された「カテゴリ」に分類する

コピーしてtest = ○○に貼り付けて実行

```
nlp = pipeline('sentiment-analysis')
text = "An exhilarating cinematic masterpiece that seamlessly blends captivating storytelling with breathtaking visuals."
nlp(text)
```

```
No model was supplied, defaulted to distilbert-base-uncased-finetuned-sst-2-english and revision af0f99b (hf)
Using a pipeline without specifying a model name and revision in production is not recommended.
[{'label': 'POSITIVE', 'score': 0.9998712539672852}]
```

魅力的と判断してPositive?

# テキスト分類：「テキスト」を事前準備された「カテゴリ」に分類する

コピーしてtest = ○○に貼り付けて実行

```
nlp = pipeline('sentiment-analysis')
text = "An exhilarating cinematic masterpiece that seamlessly blends captivating storytelling with breathtaking visuals."
nlp(text)
```

No model was supplied, defaulted to distilbert-base-uncased-finetuned-sst-2-english and revision af0f99b ([ht](#)  
Using a pipeline without specifying a model name and revision in production is not recommended.  
[{'label': 'POSITIVE', 'score': 0.9998712539672852}]

魅力的と判断してPositive?

長文の文字列は”¥”で改行できる  
(windowsはキーボードの右下)  
(macは”option”+”¥”)

```
text = "An exhilarating cinematic masterpiece that \
seamlessly blends captivating storytelling \
with breathtaking visuals."
```

# テキスト分類：「テキスト」を事前準備された「カテゴリ」に分類する

negativeも出力してみよう

あなた  
悪いレビューも英語で書いて。その後日本語にして。英語はtext="レビュー"にして。

< 2 / 2 >

ChatGPT  
英語: "A tedious and uninspiring film that fails to engage the audience with its dull plot and lackluster performances."  
日本語: "つまらなく、インスピアイアを感じさせない映画で、退屈なプロットと魅力を欠いた演技が観客を引き込むことに失敗しています。"  
   

うまくいってない  
ChatGPTは指示(プロンプト)は、分かりづらい  
指示だと正しく出力してくれません

# テキスト分類：「テキスト」を事前準備された「カテゴリ」に分類する

negativeも出力してみよう

プロンプト内の改行は(shift+enter)



あなた

悪いレビューも英語で書いて。その後日本語にして。英語はtext="レビュー"にして。

< 2 / 2 >



ChatGPT

英語: "A tedious and uninspiring film that fails to engage the audience with its dull plot and lackluster performances."

日本語: "つまらなく、インスピアイを感じさせない映画で、退屈なプロットと魅力を欠いた演技が観客を引き込むことに失敗しています。"



あなた

text = 文字列の形式で、文字列に悪い英語のレビューを 1 文で書いてください。

その後に日本語で翻訳してください。



ChatGPT

text = "This movie was absolutely terrible, with a plot that made no sense and acting that was painfully bad."

日本語: "この映画はまったく酷く、意味不明なプロットと非常にひどい演技がありました。"

うまくいってない  
ChatGPTは指示(プロンプト)は、分かりづらい  
指示だと正しく出力してくれません

プロンプトエンジニアという職種が出るほどテクニックは様々あります

# テキスト分類：「テキスト」を事前準備された「カテゴリ」に分類する

ネガティブと出力された

```
nlp = pipeline('sentiment-analysis')
text = "This movie was absolutely terrible, \
        with a plot that made no sense and acting that was painfully bad."
nlp(text)
```

```
No model was supplied, defaulted to distilbert-base-uncased-finetuned-sst-2-
Using a pipeline without specifying a model name and revision in production
[{'label': 'NEGATIVE', 'score': 0.9997949004173279}]
```

?

# テキスト分類：「テキスト」を事前準備された「カテゴリ」に分類する

## ネガティブと出力された

```
nlp = pipeline('sentiment-analysis')
text = "This movie was absolutely terrible, \
        with a plot that made no sense and acting that was painfully bad."
nlp(text)
```

```
No model was supplied, defaulted to distilbert-base-uncased-finetuned-sst-2-
Using a pipeline without specifying a model name and revision in production
[{'label': 'NEGATIVE', 'score': 0.9997949004173279}]
```

毅頭 あなた  
次の文を訳して

No model was supplied, defaulted to distilbert-base-uncased-finetuned-  
sst-2-english and revision af0f99b  
(<https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english>).  
Using a pipeline without specifying a model name and revision in  
production is not recommended.

ChatGPT

モデルが指定されていないため、distilbert-base-uncased-finetuned-sst-2-  
englishおよびリビジョンaf0f99b (<https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english>) がデフォルトで使用されました。本番環境でモデル名とリビジョンを指定せずにパイプラインを使用することは推奨されていません。

## pipelineは色んなオプションがあります

### pipeline(タスク)だけだったので、初期設定の

### "distilbert-base-uncased-finetuned-sst-2-english"

distilbertモデルベース  
(transformerの1種)

前処理で小文字化

SST-2でファインチューニング

英語版

### を使用している

# Pipeline()で指定できるタスク

- **text-classification** : テキスト分類
- **question-answering** : 質問応答
- **summarization** : 要約
- **text-generation** : テキスト生成
- **fill-mask** : 言語モデル(MLM)
- **ner** : 固有表現抽出
- **translation\_xx\_to\_yy**
- **text2text-generation** : text-to-text(翻訳、要約、質問応答)
- **feature-extraction** : 特徴抽出
- **zero-shot-classification** : Zero-Shot分類(未学習カテゴリでの分類)
- **convensational** : 対話
- など

# 自然言語処理タスク

- ・質問応答：「コンテキスト」と「質問」からコンテキスト内に含まれる「答え」を抽出する。「コンテキスト」は、文脈、前後関係、事情、背景、状況などを示すひとまとめの文章にする

コンテキスト

「自然言語処理 (NLP) は、コンピュータが人間の言語（自然言語）を理解し、生成する技術のことです。NLPは人間とコンピュータのコミュニケーションを助け、テキストデータの分析や処理を自動化します。主なタスクには、テキスト分類、機械翻訳、感情分析、文章生成、音声認識などがあります。NLPは機械学習やディープラーニングの技術を用いて、言語の複雑なパターンや構造を学習します。」

質問

自然言語処理とはなんですか

答え

コンピュータが人間の言語を理解し生成する技術

# 質問応答：「コンテキスト」と「質問」からコンテキスト内に含まれる「応答」を抽出する

質問応答のタスク名："question-answering"

```
nlp = pipeline("question-answering")
context = "説明文章(コンテキスト)"
question = "質問"
nlp(question, context)
```

## 出力結果

```
{'score': 0.9849839210510254,
'start': 123,
'end': 140,
'answer': 'Type 1 and Type 2'}
```

引数に質問(question)とコンテキスト(context)を指定する  
scoreとstart(応答の開始位置)とend(応答の終了位置)とanswerが得られる  
つまり要約しているわけではなく抜き出す形で質問に答える

このパイプラインのモデルはデータセット「SQuAD」で学習している  
SQuAD : Wikipediaの記事の内容に対する質問とその回答をクラウド  
ソーシングで作成したデータセット

The screenshot shows the SQuAD Explorer interface. At the top, there's a navigation bar with 'Home', 'Explore 2.0', and 'Explore 1.1'. The main header is 'SQuAD2.0: The Stanford Question Answering Dataset'. Below the header, there's a section titled 'What is SQuAD?' which provides a brief description of the dataset. To the right, there's a 'Leaderboard' section showing the top performing models. The first entry is 'Human Performance Stanford University' with a score of 86.831 and 89.452 F1. The second entry is 'IE-Net (ensemble) RICOH\_SRCB\_DML' with a score of 90.939 and 93.214 F1. The third entry is 'FPNet (ensemble) Ant Service Intelligence Team' with a score of 90.871 and 93.183 F1.

# 質問応答：「コンテキスト」と「質問」からコンテキスト内に含まれる「応答」を抽出する

## 説明文と質問を作ろう

あなた  
糖尿病について6行で英語で説明して。その後にその中に答えが含まれる簡単な英語の質問を作って。

ChatGPT  
**Diabetes Explanation (6 lines):**

Diabetes is a chronic health condition that affects how your body manages blood sugar (glucose). There are two main types: Type 1 and Type 2. In Type 1 diabetes, the immune system mistakenly attacks and destroys insulin-producing cells in the pancreas, leading to a lack of insulin. Type 2 diabetes, on the other hand, is often related to lifestyle factors like diet and physical activity, where the body becomes resistant to insulin. Both types result in elevated blood sugar levels, which can lead to various health complications if not properly managed.

**Q: What are the two main types of diabetes?**

A: The two main types of diabetes are Type 1 and Type 2.

日本語にして

ChatGPT  
糖尿病の説明（6行）：

糖尿病は、体が血糖（グルコース）を調節する方法に影響を与える慢性の健康状態です。主なタイプは2つあり、タイプ1とタイプ2です。タイプ1糖尿病では、免疫系が膵臓のインスリンを生成する細胞を誤って攻撃し、インスリン不足を引き起こします。一方、タイプ2糖尿病は、食事や運動などの生活習慣要因に関連し、体がインスリンに対して抵抗性を持つ状態です。どちらのタイプも血糖値が上昇し、適切に管理されない場合、さまざまな健康の合併症につながる可能性があります。

**Q: 糖尿病の主な2つのタイプは何ですか？**

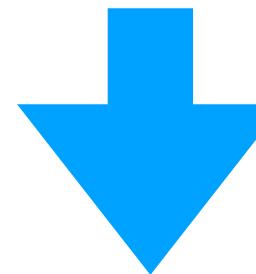
A: 糖尿病の主な2つのタイプは、タイプ1とタイプ2です。



# 質問応答：「コンテキスト」と「質問」からコンテキスト内に含まれる「応答」を抽出する

それぞれを貼り付けて実行

```
nlp = pipeline("question-answering")
context = "説明文章(コンテキスト)"
question = "質問"
nlp(question,context)
```



改行はしなくてもいいです

```
nlp = pipeline("question-answering")
context = "Diabetes is a chronic health condition that affects how your body manages blood sugar (glucose). \
There are two main types: Type 1 and Type 2. In Type 1 diabetes, the immune \
system mistakenly attacks and destroys insulin-producing cells in the pancreas, \
leading to a lack of insulin. Type 2 diabetes, on the other hand, is often related \
to lifestyle factors like diet and physical activity, where the body becomes \
resistant to insulin. Both types result in elevated blood sugar levels, \
which can lead to various health complications if not properly managed."
question = "What are the two main types of diabetes?"
nlp(question,context)
```

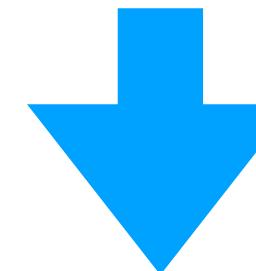
# 質問応答：「コンテキスト」と「質問」からコンテキスト内に含まれる「応答」を抽出する

それを貼り付けて実行

```
nlp = pipeline("question-answering")
context = "説明文章(コンテキスト)"
question = "質問"
nlp(question,context)
```

正解

```
{ 'score': 0.9849839210510254,
  'start': 123,
  'end': 140,
  'answer': 'Type 1 and Type 2'}
```



改行はしなくてもいいです

```
nlp = pipeline("question-answering")
context = "Diabetes is a chronic health condition that affects how your body manages blood sugar (glucose). \
There are two main types: Type 1 and Type 2. In Type 1 diabetes, the immune \
system mistakenly attacks and destroys insulin-producing cells in the pancreas, \
leading to a lack of insulin. Type 2 diabetes, on the other hand, is often related \
to lifestyle factors like diet and physical activity, where the body becomes \
resistant to insulin. Both types result in elevated blood sugar levels, \
which can lead to various health complications if not properly managed."
question = "What are the two main types of diabetes?"
nlp(question,context)
```

このタスクの初期設定のモデルは"distilbert-base-cased-distilled-squad"

# Pipeline()で指定できるタスク

- **text-classification** : テキスト分類
- **question-answering** : 質問応答
- **summarization** : 要約
- **text-generation** : テキスト生成
- **fill-mask** : 言語モデル(MLM)
- **ner** : 固有表現抽出
- **translation\_xx\_to\_yy**
- **text2text-generation** : text-to-text(翻訳、要約、質問応答)
- **feature-extraction** : 特徴抽出
- **zero-shot-classification** : Zero-Shot分類(未学習カテゴリでの分類)
- **convensational** : 対話
- など

# 自然言語処理タスク

- ・要約：「本文」(長い文章)を「要約」(短い文章)にする

## 本文

「Hugginface Transformerは、Huggingfaceが提供している、自然言語処理に特化した深層学習フレームワークです。TensorFlowとPytorchの両方に対応しています。テキスト分類、質問応答、要約などの自然言語処理タスクを深層学習を使って解くことができます。

## 要約

「Huggingface Transformerは、自然言語処理に特化した深層学習フレームワークです」

# 自然言語処理タスク

- 要約：「本文」(長い文章)を「要約」(短い文章)にする

## 本文

「Hugginface Transformerは、Huggingfaceが提供している、自然言語処理に特化した深層学習フレームワークです。TensorFlowとPytorchの両方に対応しています。テキスト分類、質問応答、要約などの自然言語処理タスクを深層学習を使って解くことができます。

## 要約

「Huggingface Transformerは、自然言語処理に特化した深層学習フレームワークです」

## 要約のタスク名：'summarization'

```
nlp = pipeline('summarization')
text = "本文"
nlp(text, max_length=最大長の長さ, min_length=最小長の長さ)
```

引数にtext, max\_length(最大長)とmin\_length(最小長)を指定する

データセット  
「CNN/DailyMailDataset」  
CNNのニュース記事のデータセット  
[https://huggingface.co/datasets/cnn\\_dailymail](https://huggingface.co/datasets/cnn_dailymail)

# 要約：「本文」(長い文章)を「要約」(短い文章)にする

あなた  
歯周病について6行で英語で説明して  


ChatGPT  
**Gingivitis Explanation (6 lines):**

Gingivitis, also known as gum disease, is a common dental condition characterized by inflammation of the gums. It is often caused by poor oral hygiene practices, leading to the buildup of plaque on teeth. Common symptoms include redness, swelling, and bleeding of the gums, especially during brushing or flossing. Gingivitis is a mild and reversible form of gum disease, but if left untreated, it can progress to a more severe condition called periodontitis, which can result in tooth loss and other complications. Regular dental check-ups and good oral hygiene are essential in preventing and managing gingivitis.

あなた  
日本語にして  


ChatGPT  
**歯周病の説明 (6行) :**

歯周病、またの名を歯肉疾患とも呼ばれ、歯肉の炎症が特徴の一般的な歯科疾患です。これは通常、不適切な口腔衛生習慣により、歯にプラークがたまることに起因します。一般的な症状には、歯ぐきの赤み、腫れ、歯ぐきからの出血が含まれます、特に歯磨きやフロスをする際に。歯周病は軽度で逆転可能な歯肉疾患ですが、治療を受けない場合、さらに重度の歯周病である歯周炎に進行し、歯の喪失やその他の合併症を引き起こす可能性があります。歯周病を予防し管理するためには、定期的な歯科検診と良好な口腔衛生が不可欠です。

# 要約：「本文」(長い文章)を「要約」(短い文章)にする

あなた  
歯周病について6行で英語で説明して

ChatGPT  
**Gingivitis Explanation (6 lines):**

Gingivitis, also known as gum disease, is a common dental condition characterized by inflammation of the gums. It is often caused by poor oral hygiene practices, leading to the buildup of plaque on teeth. Common symptoms include redness, swelling, and bleeding of the gums, especially during brushing or flossing. Gingivitis is a mild and reversible form of gum disease, but if left untreated, it can progress to a more severe condition called periodontitis, which can result in tooth loss and other complications. Regular dental check-ups and good oral hygiene are essential in preventing and managing gingivitis.

あなた  
日本語にして

ChatGPT  
**歯周病の説明 (6行) :**

歯周病、またの名を歯肉疾患とも呼ばれ、歯肉の炎症が特徴の一般的な歯科疾患です。これは通常、不適切な口腔衛生習慣により、歯にプラークがたまることに起因します。一般的な症状には、歯ぐきの赤み、腫れ、歯ぐきからの出血が含まれます、特に歯磨きやフロスをする際に。歯周病は軽度で逆転可能な歯肉疾患ですが、治療を受けない場合、さらに重度の歯周病である歯周炎に進行し、歯の喪失やその他の合併症を引き起こす可能性があります。歯周病を予防し管理するためには、定期的な歯科検診と良好な口腔衛生が不可欠です。

```
nlp = pipeline('summarization')
text = "Gingivitis, also known as gum disease, is a common dental condition characterized by inflammation of the
nlp(text, max_length=25,min_length=10)
```

No model was supplied, defaulted to sshleifer/distilbart-cnn-12-6 and revision a4f8f3e (<https://huggingface.co/ss>)
Using a pipeline without specifying a model name and revision in production is not recommended.

```
[{'summary_text': ' Gingivitis is a mild and reversible form of gum disease . It is often caused by poor oral
hygiene practices'}]
```

歯肉炎は、軽度で逆転可能な歯肉疾患の一形態です。

これは通常、不適切な口腔衛生習慣によって引き起こされることが多いです。

大体あってる(が、少し違う)

# それぞれのモデルについて

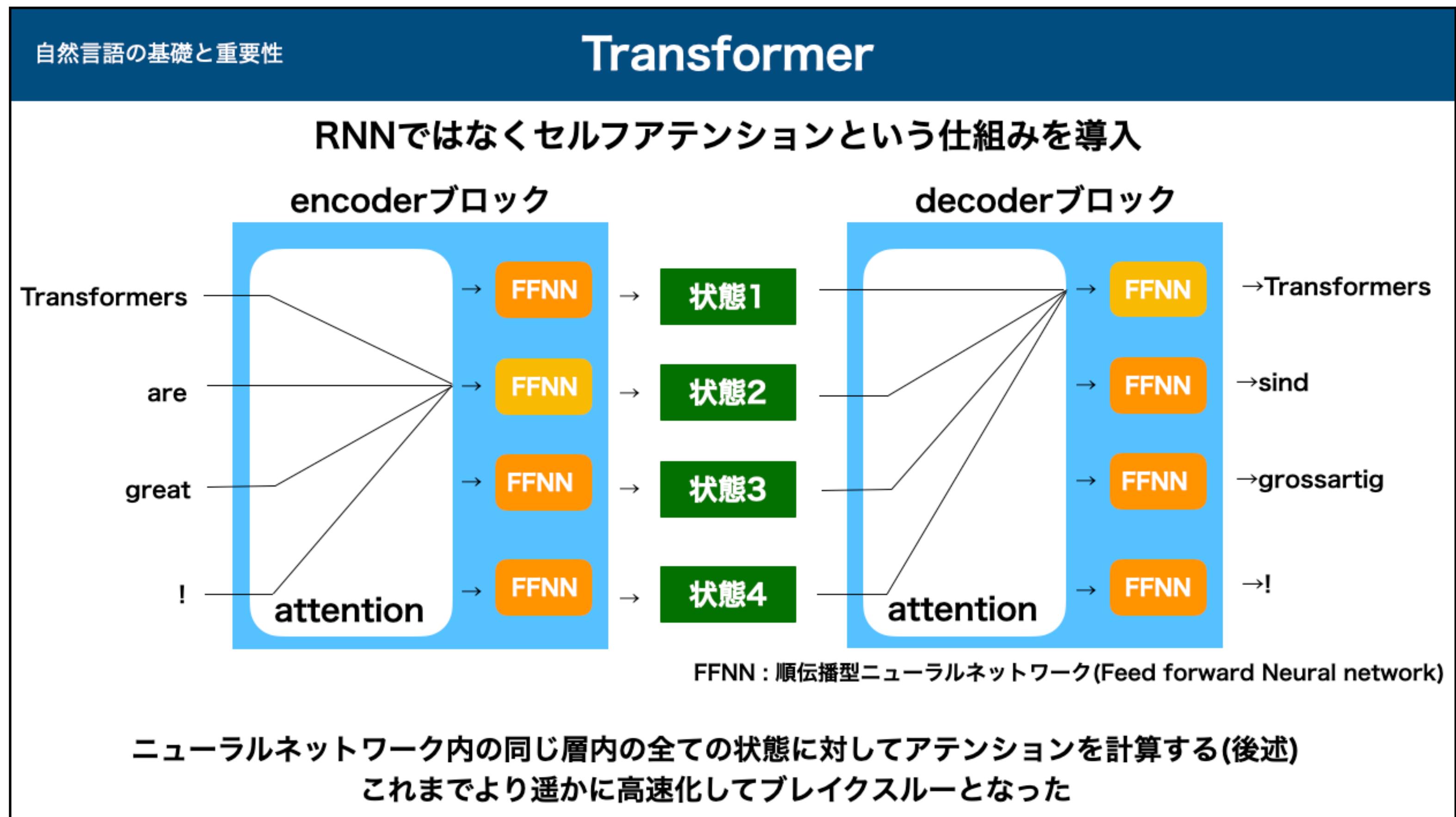
- **text-classification** : テキスト分類  
distilbert-base-uncased-finetuned-sst-2-english
  - **question-answering** : 質問応答
  - **summarization** : 要約  
distilbert-base-cased-distilled-squad
- sshleifer/distilbart-cnn-12-6

distilbert/distilbartとは何か？

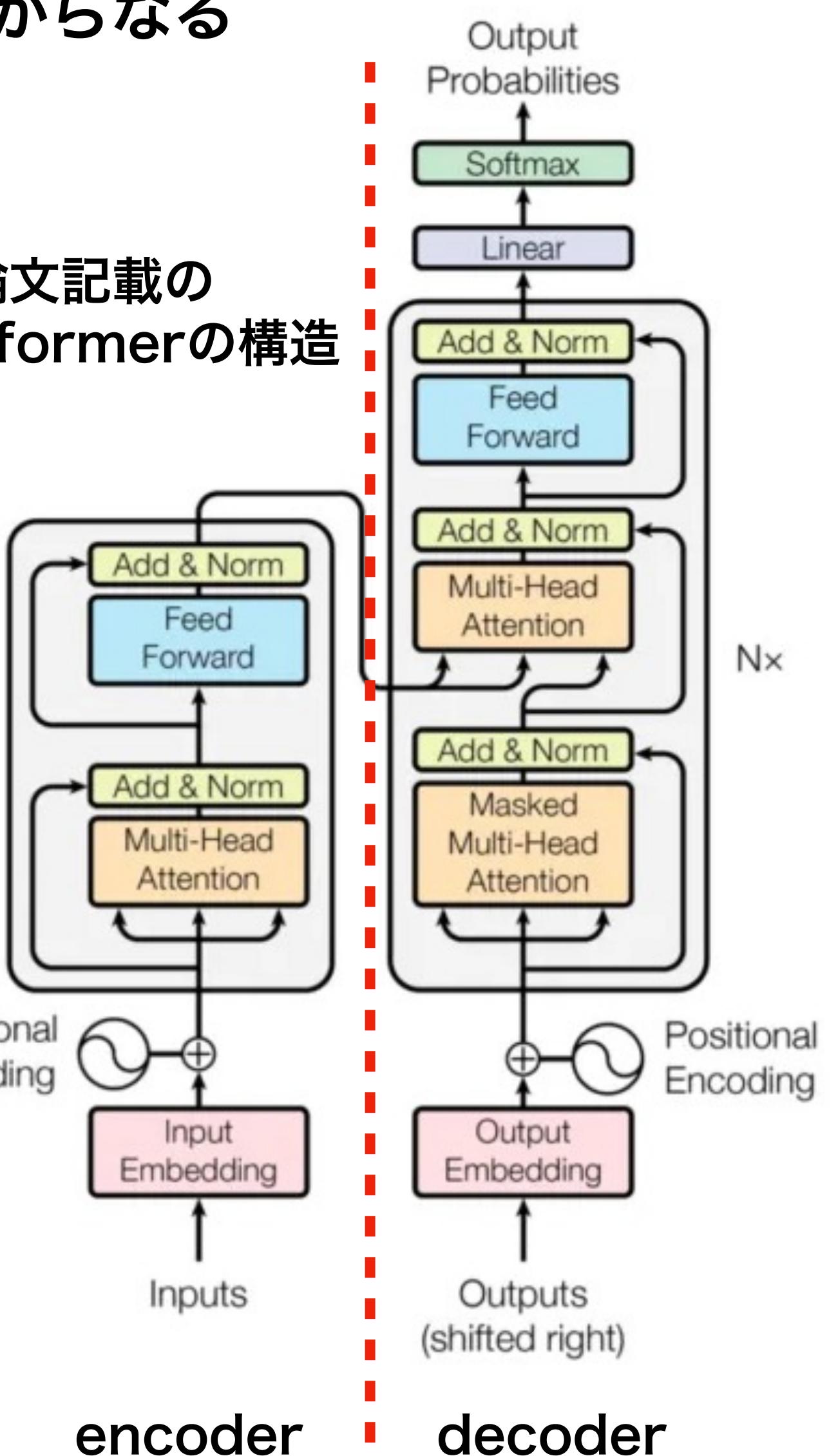
Bertモデルの蒸留/Bartモデルの蒸留

# Transformerの原型

オリジナルのTransformerはencoder(圧縮)とdecoder(生成)の2つのブロックからなる



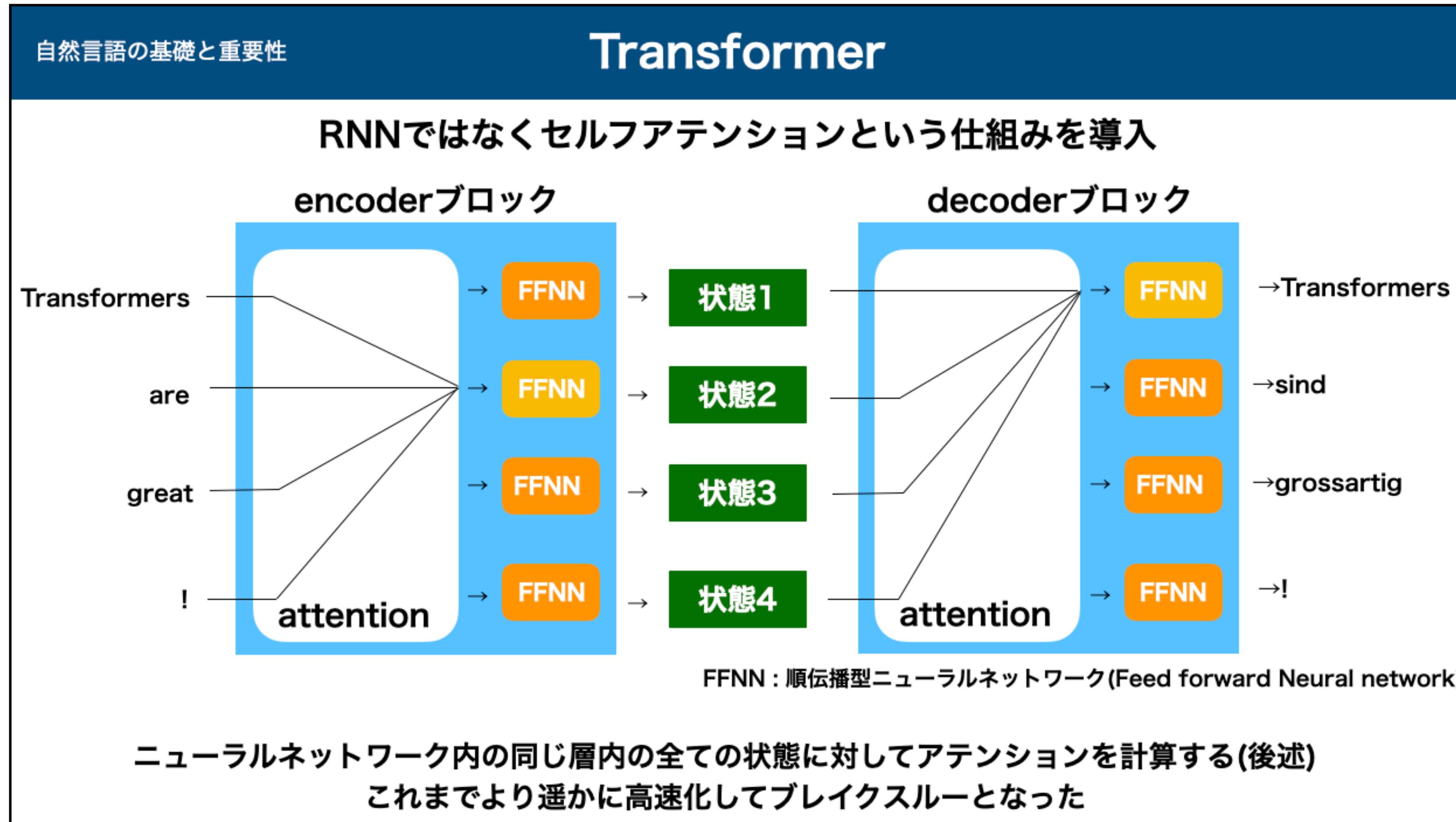
論文記載の  
Transformerの構造



BertやBart、GPT(ChatGPT)はこれらの派生系

# Transformerの原型

オリジナルのTransformerはencoder(圧縮)とdecoder(生成)の2つのブロックからなる



Encoder(圧縮)のみ  
(オートエンコーディングモデル)

Bert, ALBERT, RoBERTaなど

decoder(生成)のみ  
(自己回帰モデル)

GPT系, CTRL, Reformerなど

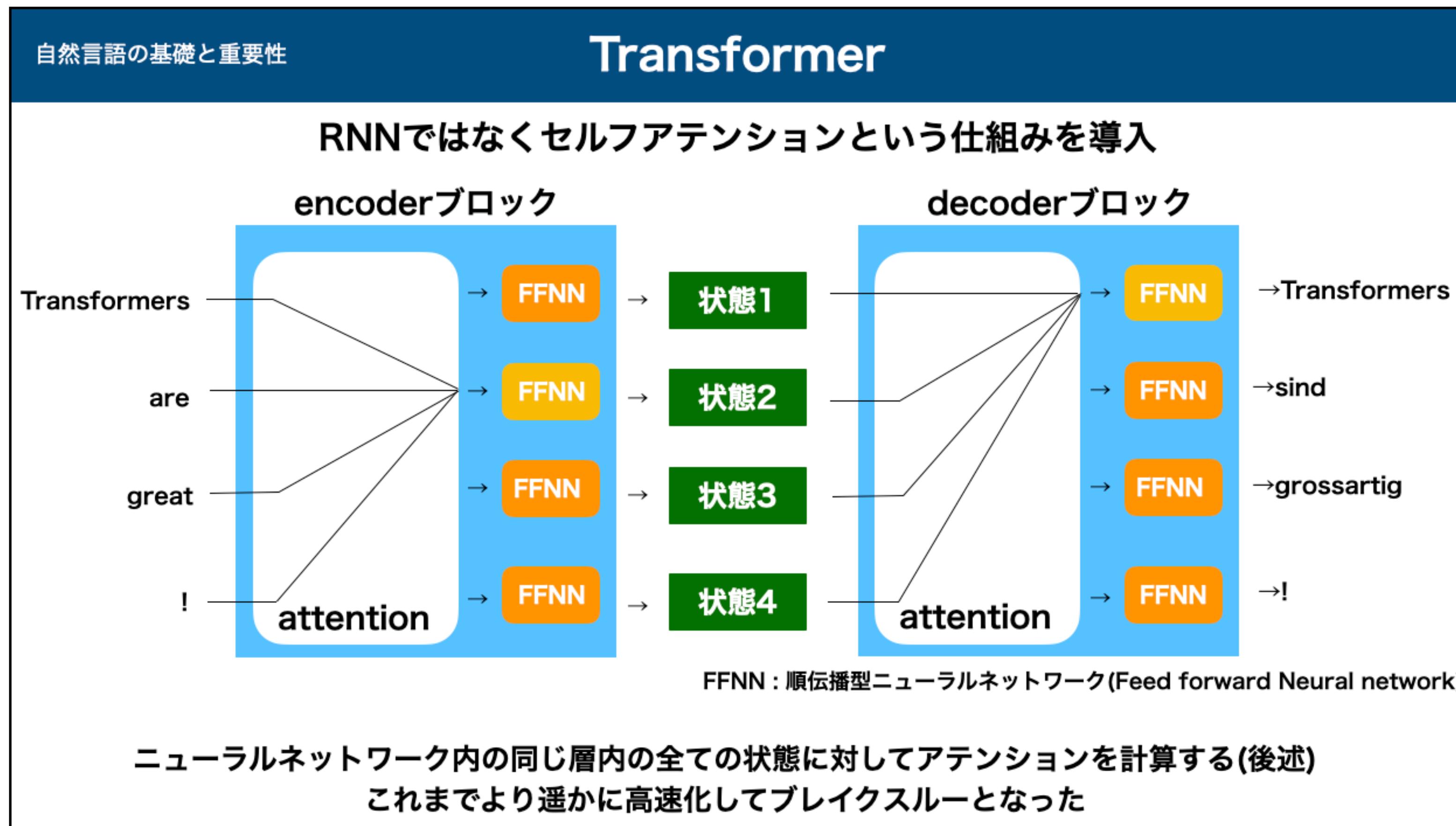
encoderとdecoder  
(Sequence-to-Sequenceモデル)

Bart , Pegasus, T5など

BertやBart、GPT(ChatGPT)はこれらの派生系

# Transformerの原型

オリジナルのTransformerはencoder(圧縮)とdecoder(生成)の2つのブロックからなる



Encoder(圧縮)のみ  
(オートエンコーディングモデル)

Bert, ALBERT, RoBERTaなど

得意なタスク : テキスト分類

decoder(生成)のみ  
(自己回帰モデル)

GPT系, CTRL, Reformerなど

得意なタスク : テキスト生成

encoderとdecoder両方  
(Sequence-to-Sequenceモデル)

Bart , Pegasus, T5など

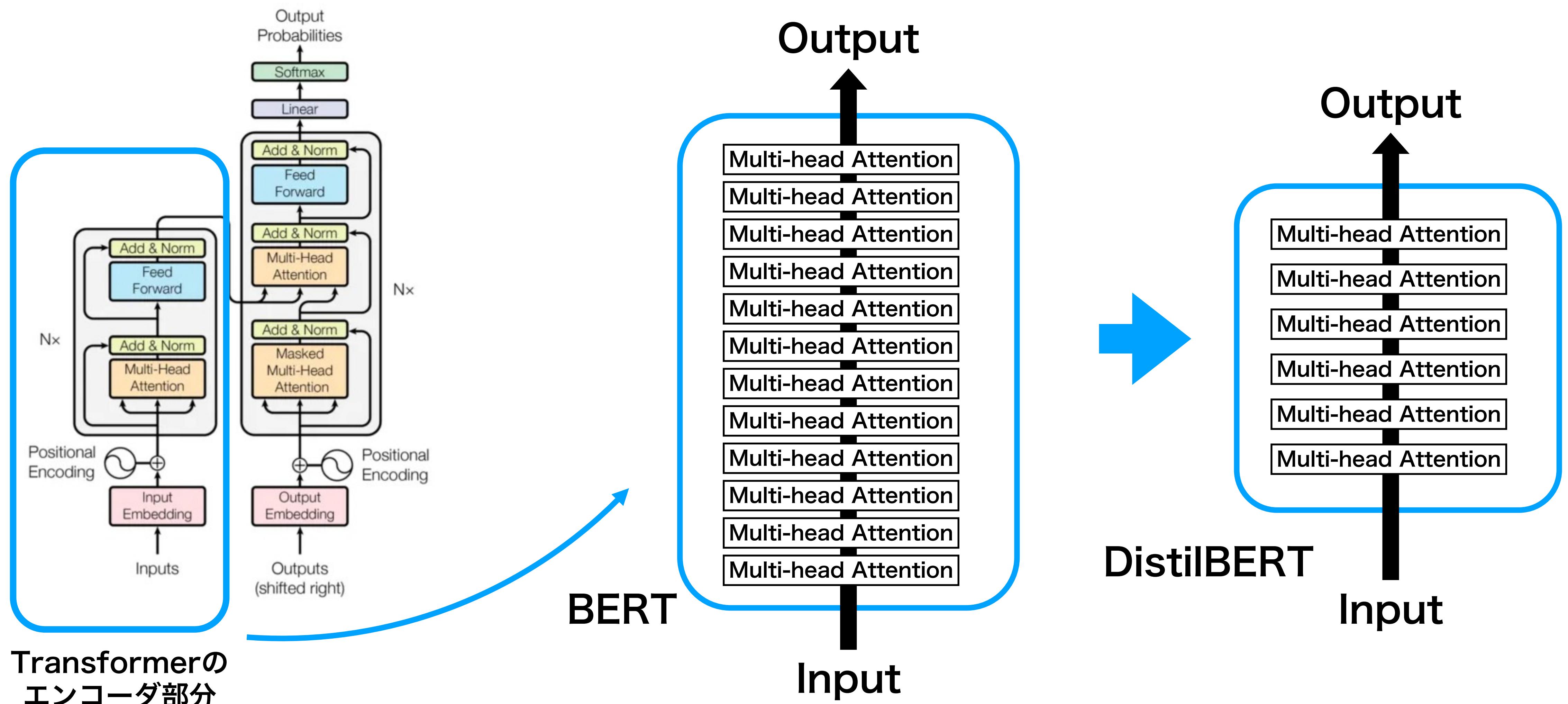
得意なタスク : 翻訳、要約、質問応答

BertやBart、GPT(ChatGPT)はこれらの派生系

# Distillation(蒸留)

BERTは左のエンコーダ部分を12回繰り返したモデル

DistilBERTは元の巨大なモデルを能力を極力落とさずに小型化したモデル



# Pipeline()で指定できるタスク

- **text-classification** : テキスト分類
  - **question-answering** : 質問応答
  - **summarization** : 要約
  - **text-generation** : テキスト生成
  - **fill-mask** : 言語モデル(MLM)
  - **ner** : 固有表現抽出
  - **translation\_xx\_to\_yy**
  - **text2text-generation** : text-to-text(翻訳、要約、質問応答)
  - **feature-extraction** : 特徴抽出
  - **zero-shot-classification** : Zero-Shot分類(未学習カテゴリでの分類)
  - **convensational** : 対話
- など

decoder(生成)のみ  
(自己回帰モデル)  
GPT系, CTRL, Reformerなど  
得意なタスク : テキスト生成

# 自然言語処理タスク

- ・テキスト生成：与えられた「テキスト」に続く、もっともらしい「テキスト」を自動生成する

テキスト 「かつて、遠い森の中に小さな熊がひっそりと住んでいました。」

生成テキスト 「彼の名前はユーリ。ユーリはこの森で生まれ育ち、森の生活を心から愛していました。彼の毎日は、木々の間を駆け回り、川で魚を捕まえ、森の美しい音に耳を傾けることで満たされていました。しかし、ある日、ユーリの平和な生活は突然変わりました。」(ChatGPTより)

## テキスト生成のタスク名：'text-generation'

```
nlp = pipeline("text-generation")
text = "I will now explain the causes of cancer."
nlp(text, max_length=200, min_length=50)
```

### データセット

WebText(40GB)

大量(800万)のWebページ上の  
テキストをデータ化したもの

引数にtext, max\_length(最大長)とmin\_length(最小長)を指定する

# テキスト生成：与えられた「テキスト」に続く、もっともらしい「テキスト」を自動生成する

"今からがんの原因について説明します"

```
nlp = pipeline("text-generation")
text = "I will now explain the causes of cancer."
nlp(text, max_length=200, min_length=50)
```

モデルは**GPT2**

```
No model was supplied, defaulted to gpt2 and revision 6c0e608 (https://huggingface.co/gpt2).
Using a pipeline without specifying a model name and revision in production is not recommended.
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
[{'generated_text': 'I will now explain the causes of cancer. He mentioned the fact that I am obese, so he was a little upset, so I said, "Listen to me," and he laughed, and he said, "Thank you." Which turned out it was just a joke. And that was the final nail in the coffin.\n\nI would not have written off the fact that I was a fat, thin, beautiful, and amazing person had my diagnosis been better. If I could have been a little better, I would. It was not going to get any easier. The last thing I wanted was to die.\n\nThe problem with the body\'s metabolism was there was little sense of time passed. The body did not work that way. My family and friends would be sick by the time there was less time left to work. So when my parents came down to visit me, I was like, "How\'re they going to eat after I die?" That was when I started to feel'}]
```

「がんの原因について説明します。彼は私が肥満であるという事実を言及したので、彼は少し不安そうで、私は『聞いてごらん』と言った。そして彼は笑い、『ありがとう』と言った。結局、それはただの冗談だった。それが最後の一撃だった。

私はもし私の診断が良ければ、私が太っていることを諦めたり、美しい素晴らしい人であることを書き消すことはなかっただろう。もしもう少し良くなれたなら、それはしたでしょう。事態は容易にはなりませんでした。私が望んでいた最後のことは死ぬことではありませんでした。体の代謝の問題は、経過した時間を感じることがほとんどありませんでした。体はそのように動作しませんでした。家族や友達は、働く時間が少なくなるまでに病気になってしまうだろう。だから両親が私を訪ねてきたとき、私は『私が死んだ後、どうやって食べるんだろう?』と思ったのです。それが私が感じ始めたときでした。」

いまいち

# テキスト生成：与えられた「テキスト」に続く、もっともらしい「テキスト」を自動生成する

“今からがんの原因について説明をします。がんの原因は”

```
nlp = pipeline("text-generation")
text = "I will now explain the causes of cancer. The causes of cancer is"
nlp(text, max_length=200, min_length=50)
```

```
No model was supplied, defaulted to gpt2 and revision 6c0e608 (https://huggingface.co/gpt2).
Using a pipeline without specifying a model name and revision in production is not recommended.
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
[{'generated_text': "I will now explain the causes of cancer. The causes of cancer is known as inflammation. Since
inflammation is one component of inflammation, we need to look at why inflammation is a cause for cancer.\n\nWhy Are
Cancer Causes of Cancer Different?\n\nThe immune system (in the eye) is the organ responsible for most of the
inflammatory molecules that cause cancer, which also include many different types of immune-modifying
substances.\n\nBecause you can fight disease by fighting or dying, cancer is one of the most common types of cancer that
cause many things in the body. It spreads into the bloodstream and is often the last thing you want.\n\nAn example of a
cancer cancer:\n\nA recent study showed the incidence of ovarian cancer and ovarian cancer are all among the more common
types of cancers in women.\n\nAnother new study found that some types of cancers were linked to multiple sclerosis, and
had an increase in the proportion of cancers that could cause cancer.\n\nThere are'}]
```

誘導してあげる  
と少し精度が向上

「がんの原因について説明します。がんの原因是炎症として知られています。炎症は炎症の要素の一部であるため、なぜ炎症ががんの原因であるのかを見ていく必要があります。なぜがんの原因是異なるのか？免疫系（目の中の）はがんを引き起こす多くの炎症分子の責任を持つ器官であり、これには多くの異なるタイプの免疫調整物質も含まれています。病気と戦ったり死んだりすることで、体内で多くのことを引き起こす最も一般的ながんの一つであるがんは、体内に多くのものを引き起こします。これは血流に広がり、通常は望ましくない最後のことです。がんの一例：最近の研究によれば、卵巣がんと卵巣がんの発生率は、女性の中でより一般的ながんのタイプに含まれています。別の新しい研究では、いくつかのがんのタイプが多発性硬化症と関連しており、がんを引き起こす可能性があるがんの割合が増加していることが示されました。」

【注意】元のテキストに誤りが含まれているため、意味の整合性が保たれていない部分があります。こちらの翻訳はそのままの内容を保持しています。

# テキスト生成：与えられた「テキスト」に続く、もっともらしい「テキスト」を自動生成する

## GPT3.5

毅顕

あなた

以下の続きの文章を作成してください。

"今からがんの原因について説明をします。がんの原因是"



ChatGPT

もちろん、続きを文章を作成します。

"今からがんの原因について説明をします。がんの原因是、細胞の異常な成長と分裂に関連しています。通常、細胞は特定のサイクルで成長し、古い細胞が新しい細胞に置き換わります。しかし、がんはこのプロセスが制御を失った結果、異常な細胞が異常な速さで増殖する状態です。この異常な成長は、細胞内の遺伝子の変異や、環境要因、遺伝的要因などによって引き起こされることがあります。さらに詳しく説明します。"



## GPT4

毅顕

あなた

以下の続きの文章を作成してください。

"今からがんの原因について説明をします。がんの原因是"



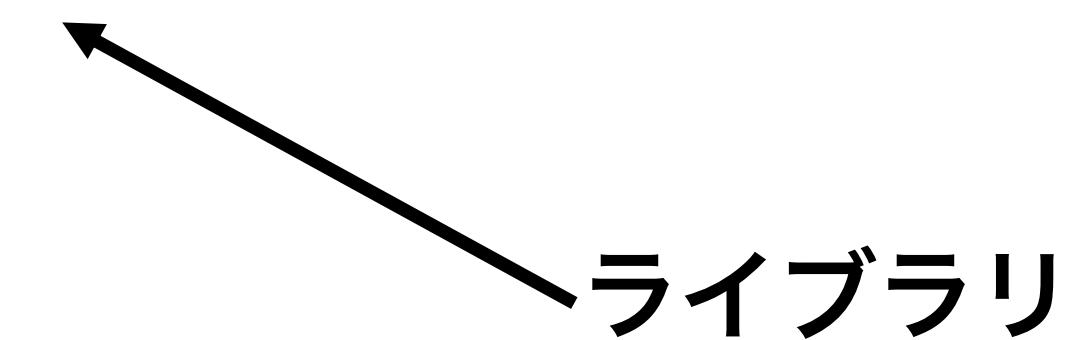
ChatGPT

"今からがんの原因について説明をします。がんの原因是、一般的には遺伝的要因、生活習慣、環境要因の三つに大別されます。まず遺伝的要因については、家族歴や特定の遺伝子変異ががんのリスクを高めることが知られています。次に生活習慣については、喫煙、不健康な食生活、運動不足などががんのリスクを増加させる要因として挙げられます。また、紫外線への長時間の露出や有害な化学物質への露出も、がんの発症に影響を及ぼすことがあります。しかし、これらの要因がすべてのがんの原因となるわけではなく、場合によっては完全には理解されていない要因も存在します。がんは多因子性の疾患であり、個々のリスク要因の影響の程度は人によって異なります。したがって、リスクを減らすためには、健康的な生活習慣を維持し、定期的な健康診断を受けることが重要です。"



# まとめ

自然言語処理の推論(予測)をTransformersのpipelineを使って実践しました

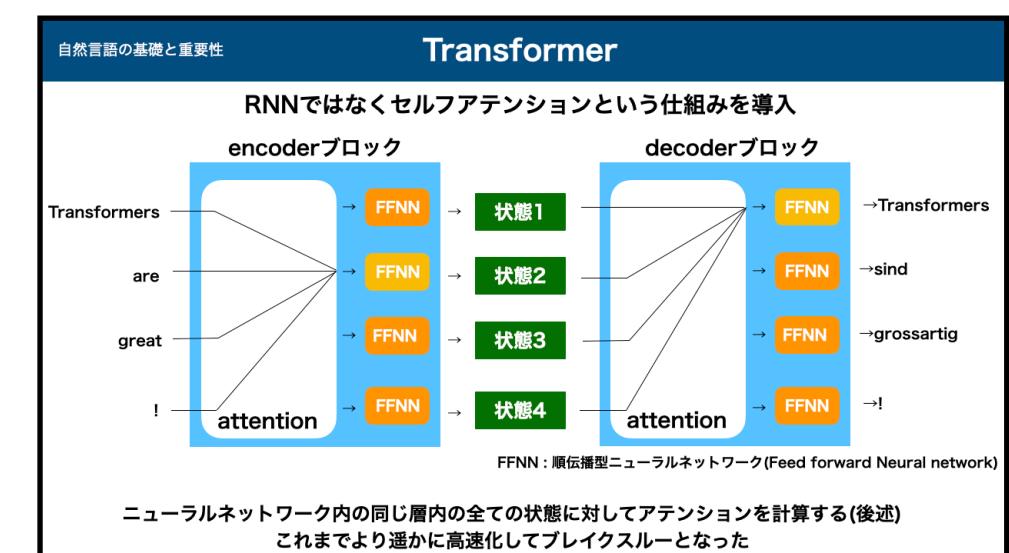


ChatGPTやこれらのAIモデルの基盤となるのは  
Transformerアーキテクチャです



BERTはTransformerのEncoder部分(圧縮)、  
GPTはTransformerのDecoder部分(生成)から構成されています

次回はpipelineではないもう一つのトークナイザーを使用します



# 課題

WebClassにある課題12をやりましょう

締め切りは1週間後の7/11の23:59です。  
締切を過ぎた課題は受け取らないので注意してください。  
(1週間後に正解をアップします)