自然言語処理 —Attention, Transformer —

https://satoyoshiharu.github.io/nlp/

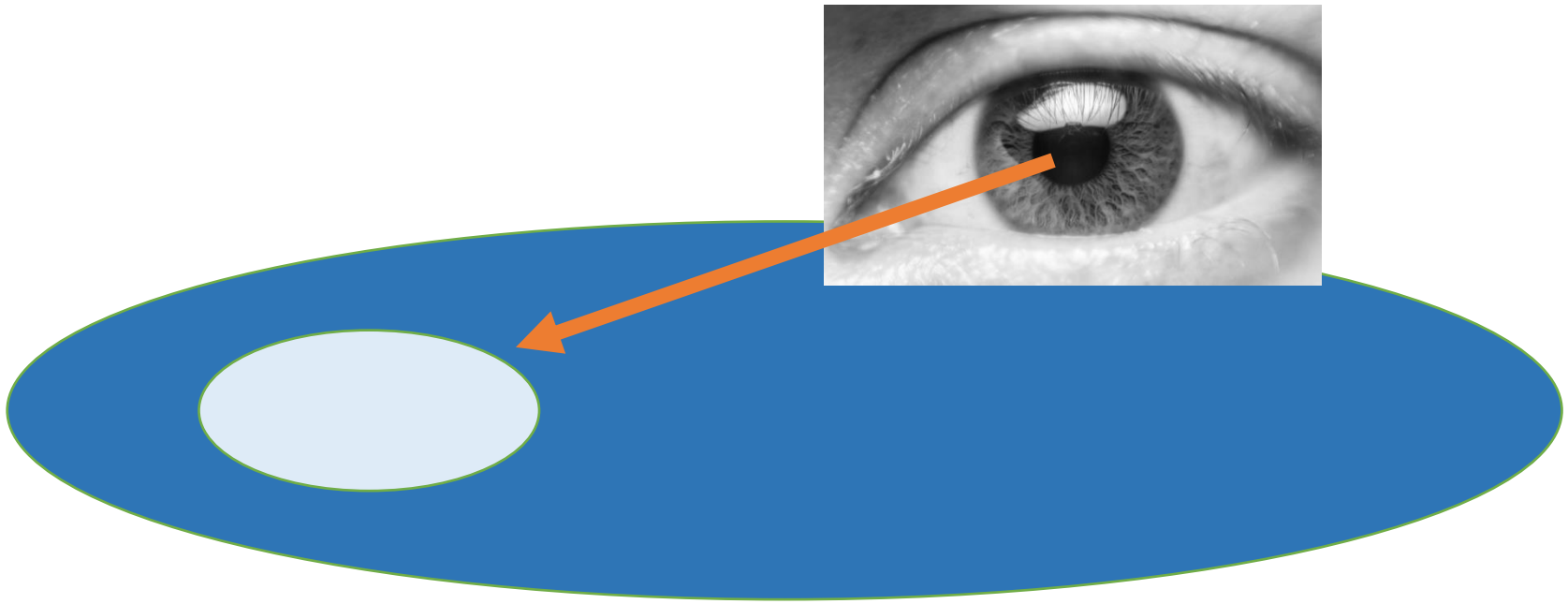
Attention、Transformerの位置づけ

- 1000本ノック第9章では、最後の一問のみTransformerネタです。
- Attentionは、自然言語処理のみならず画像処理、信号処理など他分野へ影響を及ぼしている。
- Transformerは、Attentionをもとにしている。そして、自然言語テキストの処理は、Transformerを使った巨大なシステムが続々と生まれている。

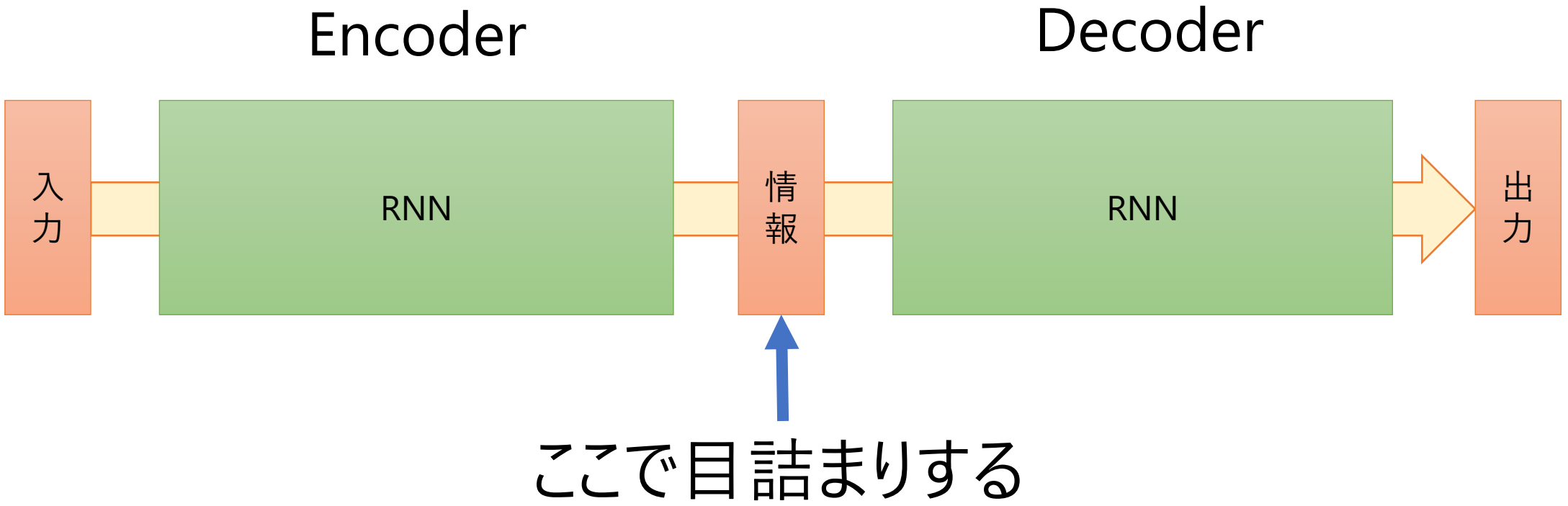
自然言語処理
Attention

解説動画

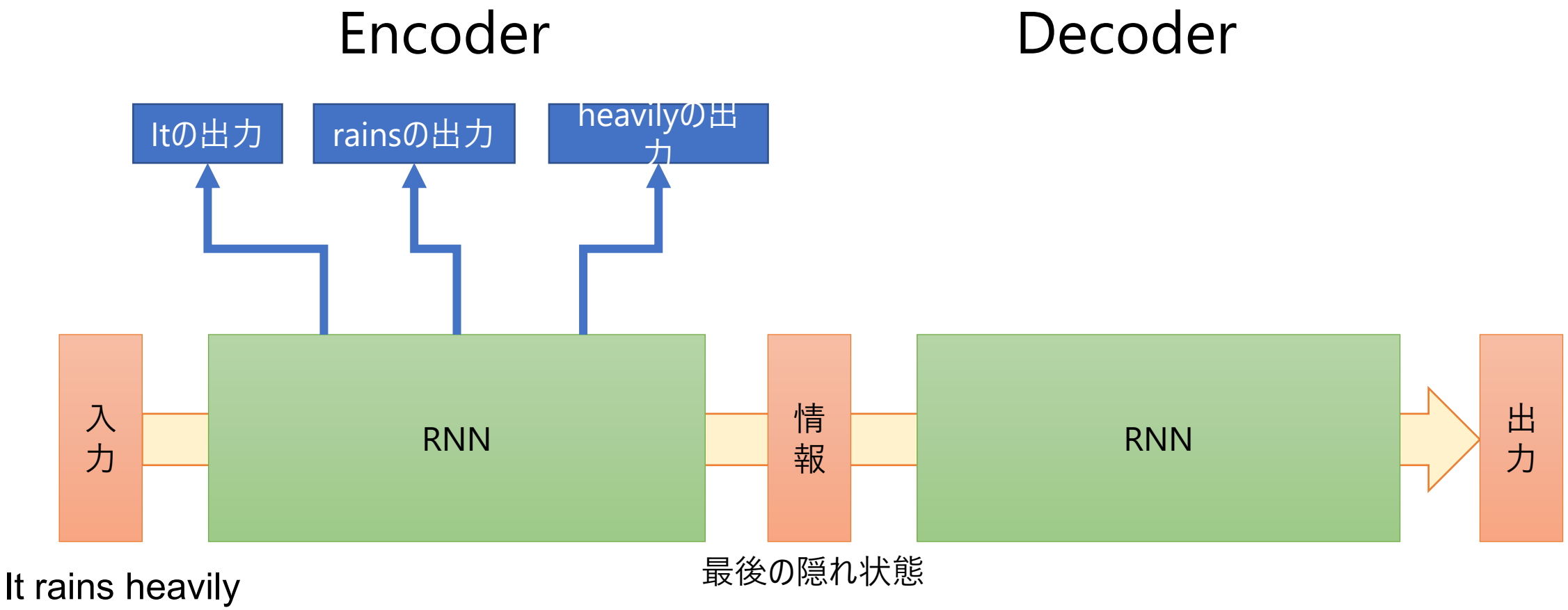
Attentionとは、



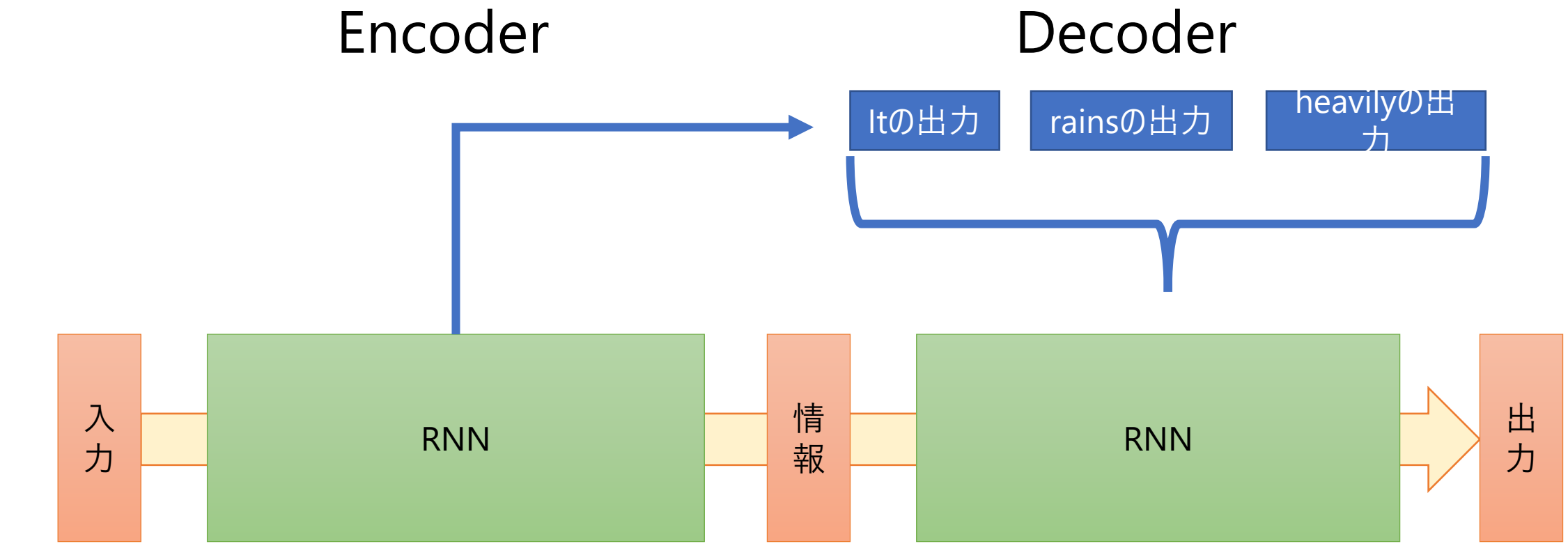
Seq2Seq：RNNへの入力が長くなると、



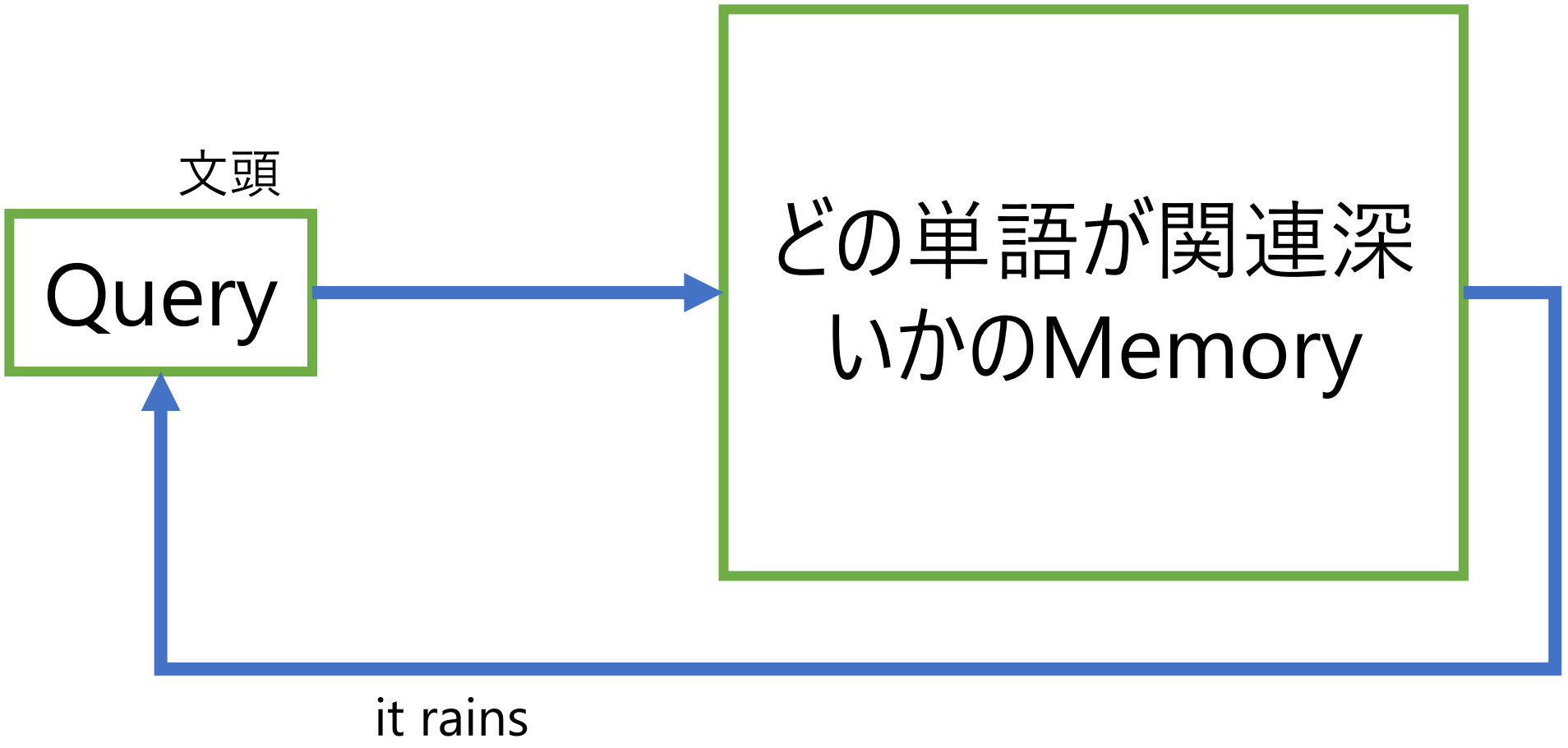
エンコーダーの各時刻の出力も利用したい



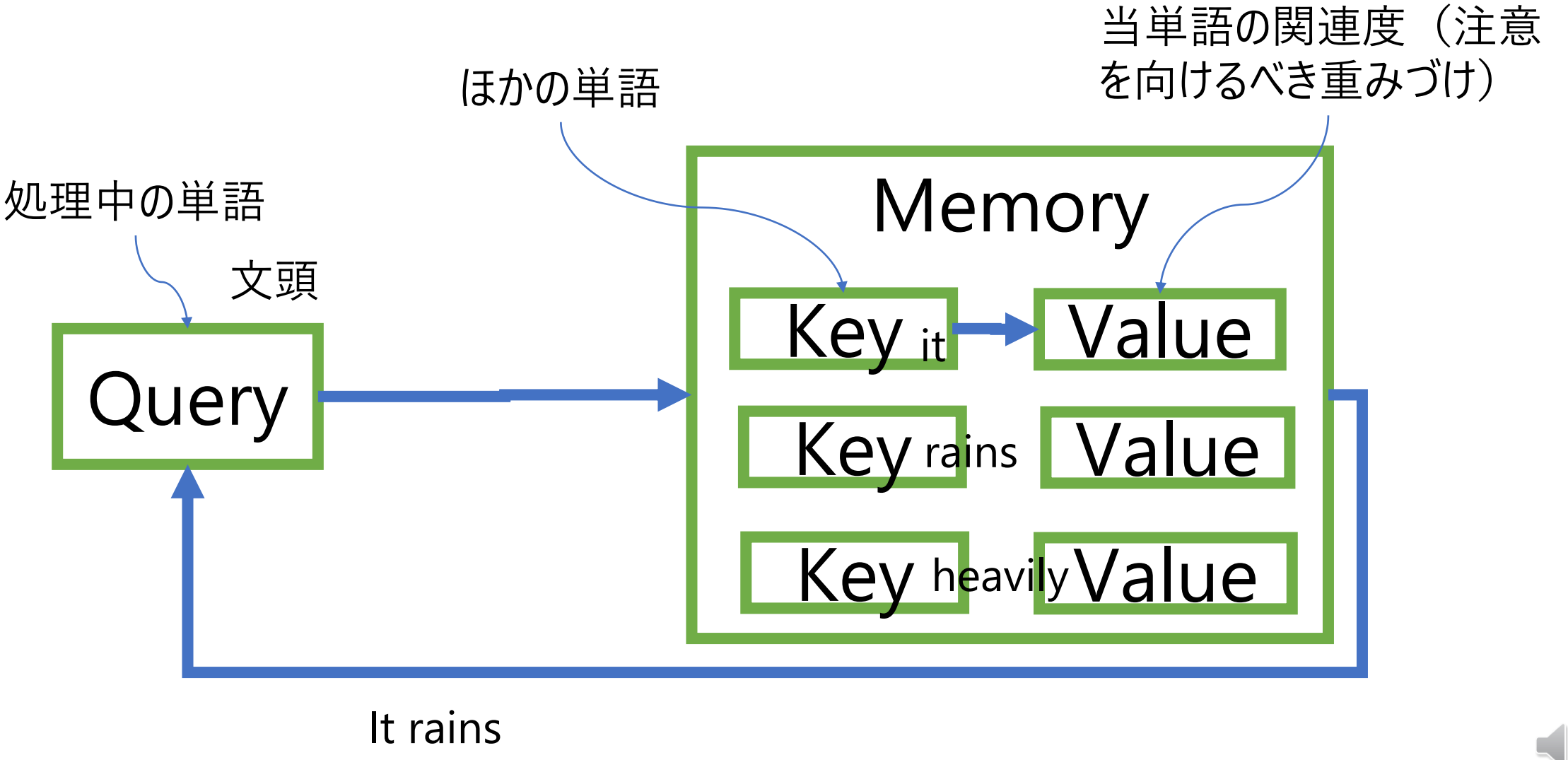
エンコーダーの出力は、1個づつ利用するのでなく、全部グローバルにみる



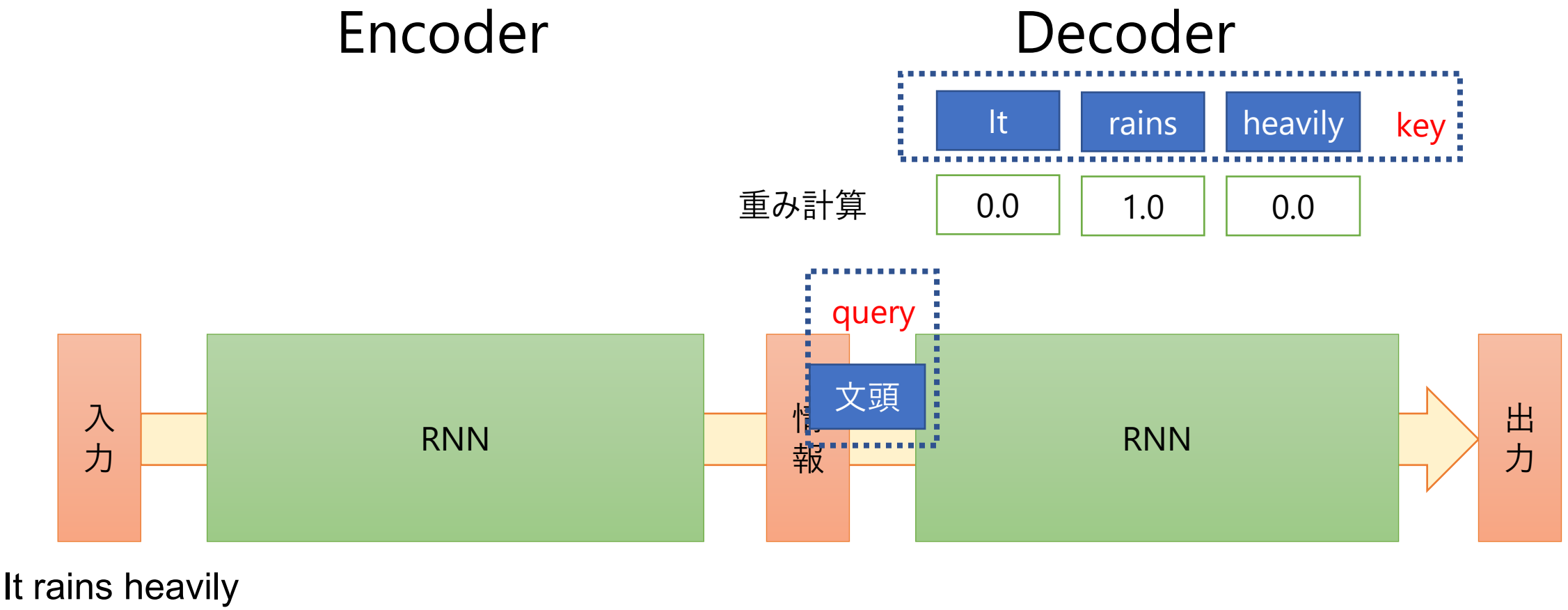
Attention = Query, Memory



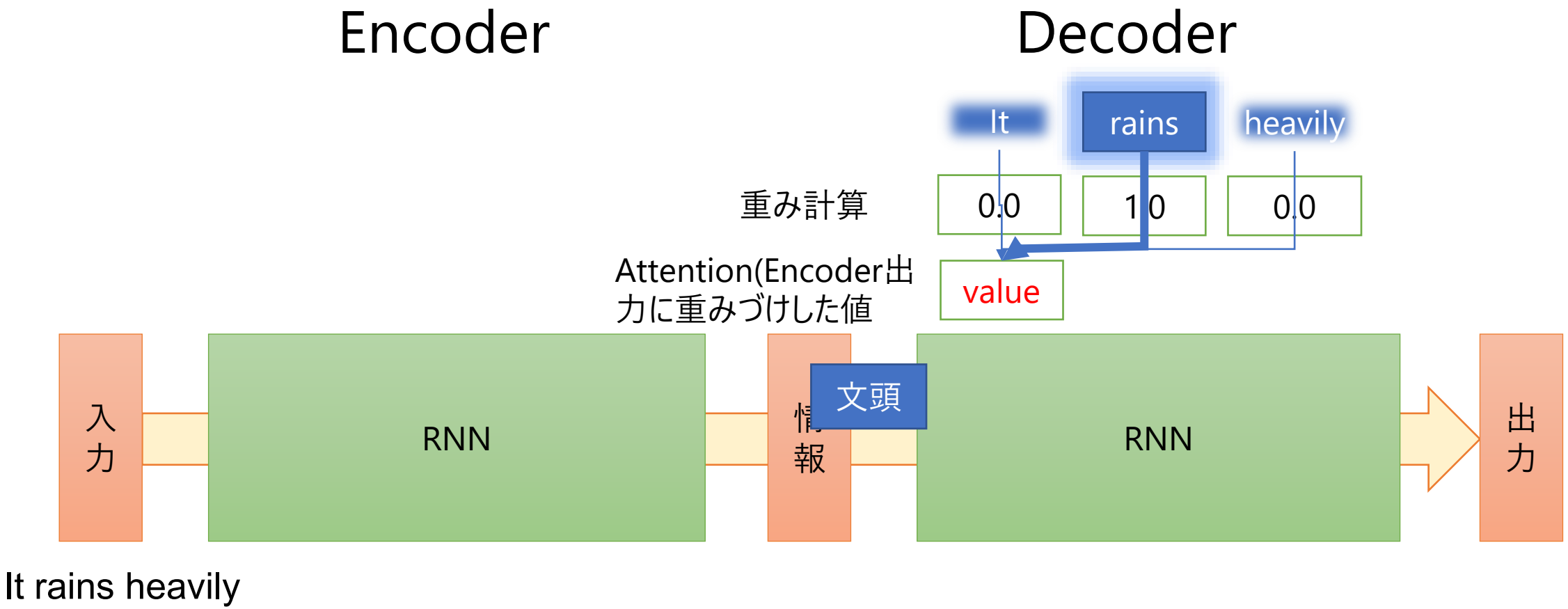
Attention = Query, Key, Value



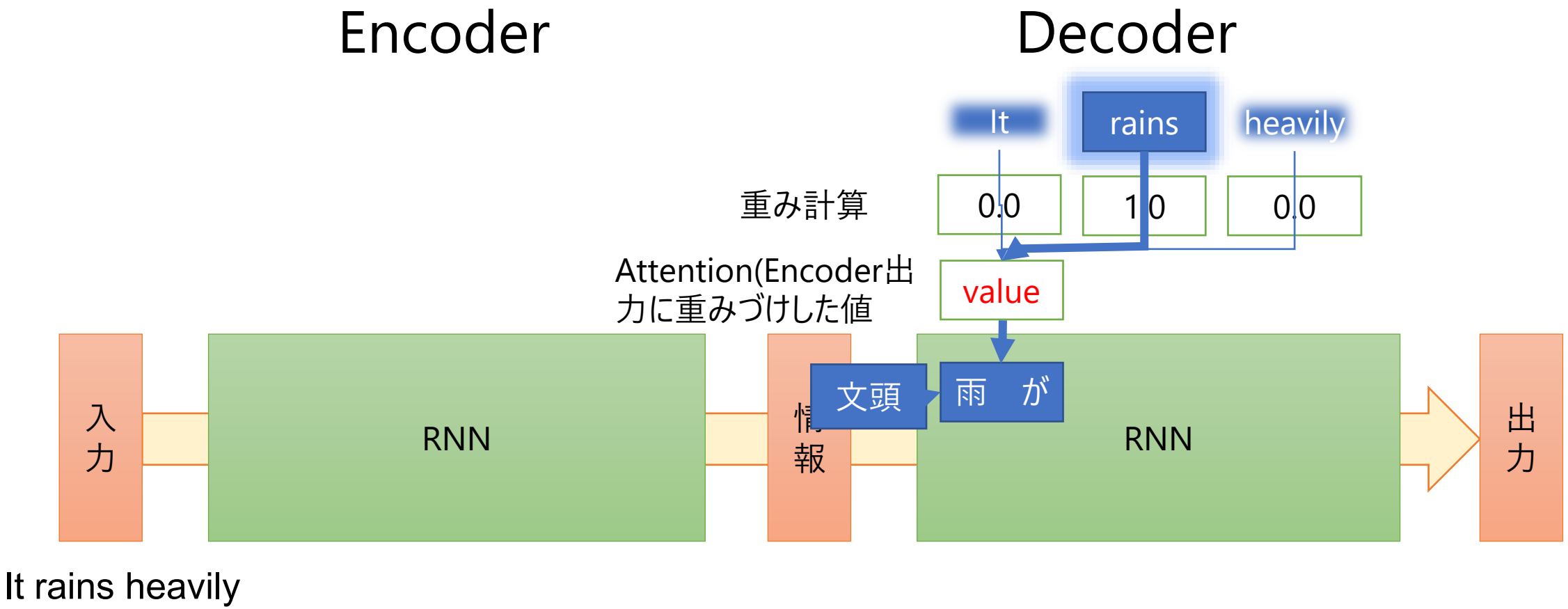
エンコーダーの出力全体のうち、特定部分に注目する



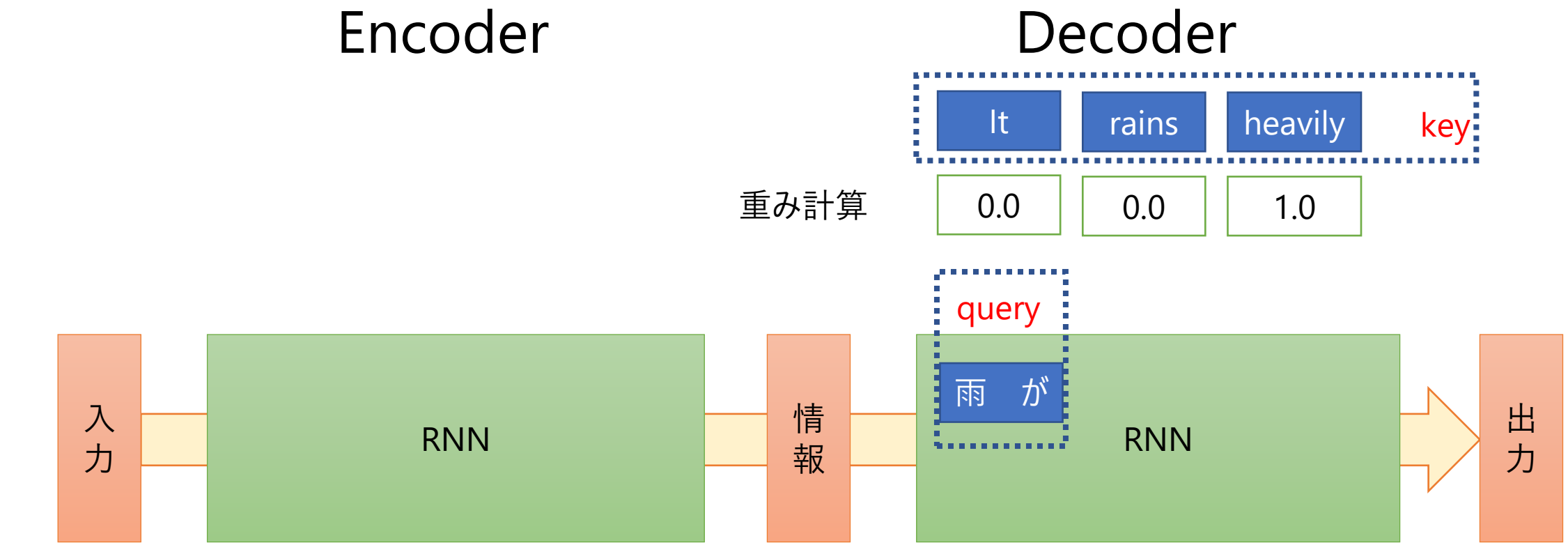
エンコーダーの出力全体のうち、特定部分に注目する



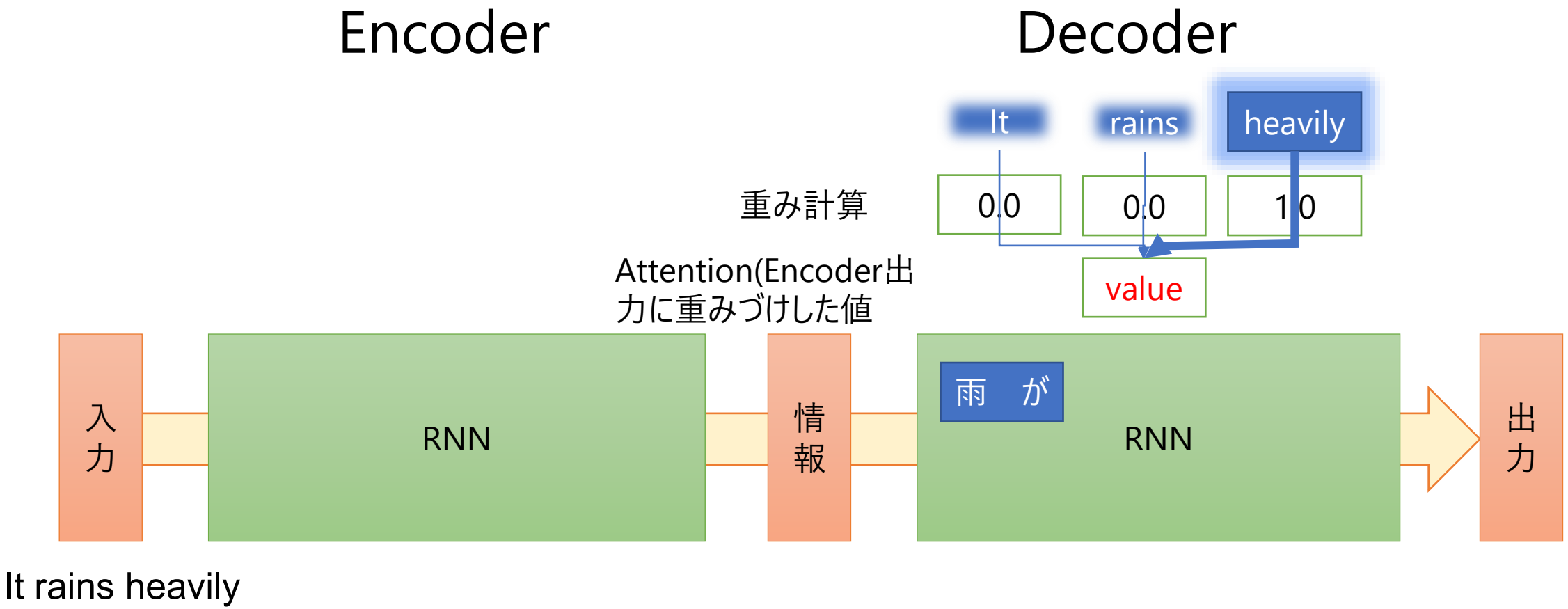
エンコーダーの出力全体のうち、特定部分に注目する



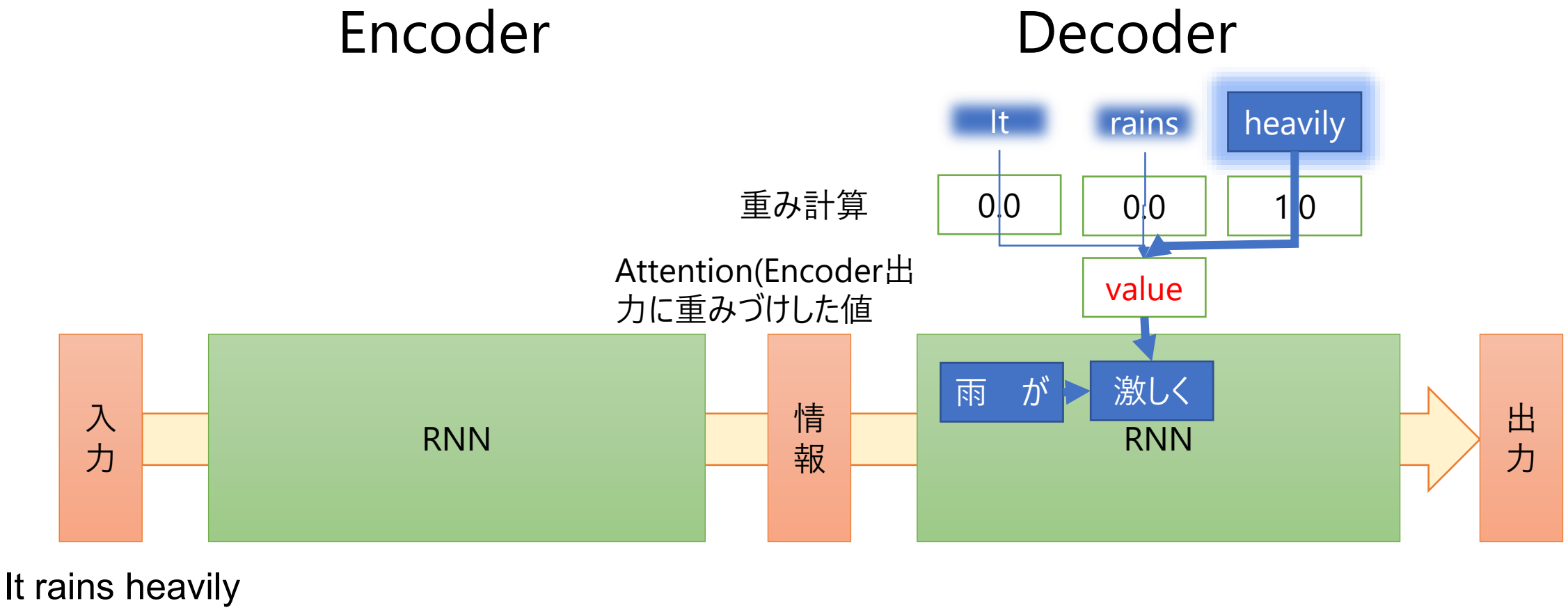
エンコーダーの出力全体のうち、特定部分に注目する



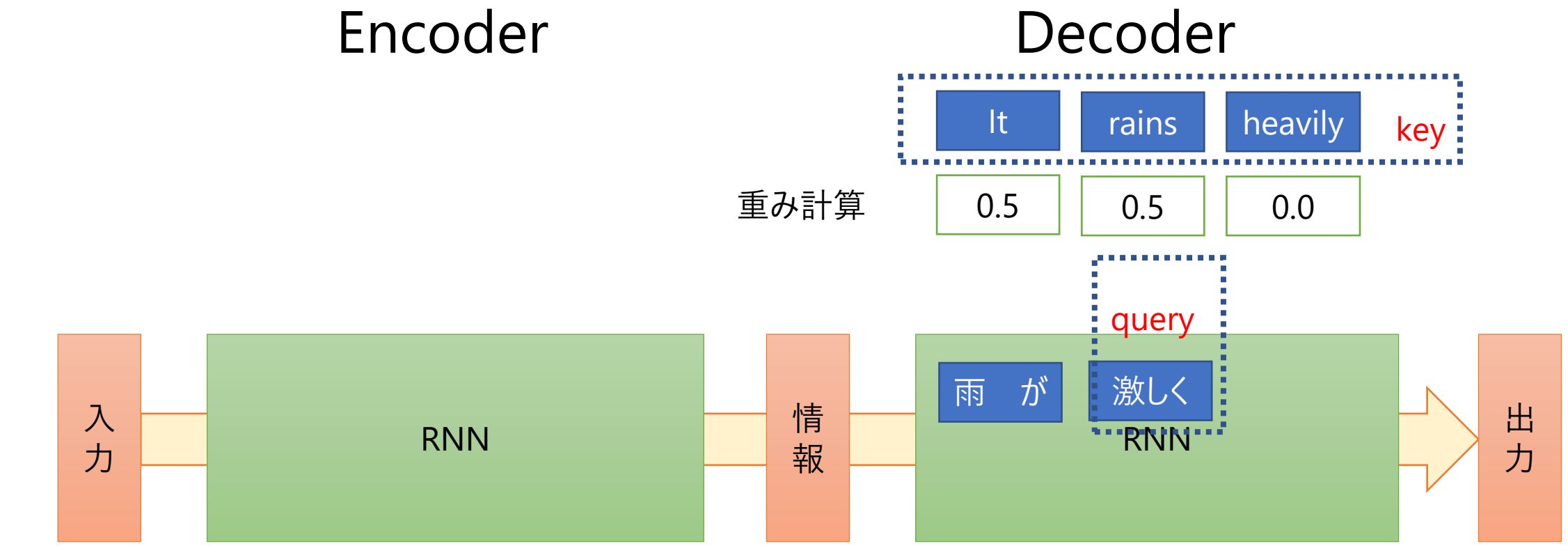
エンコーダーの出力全体のうち、特定部分に注目する



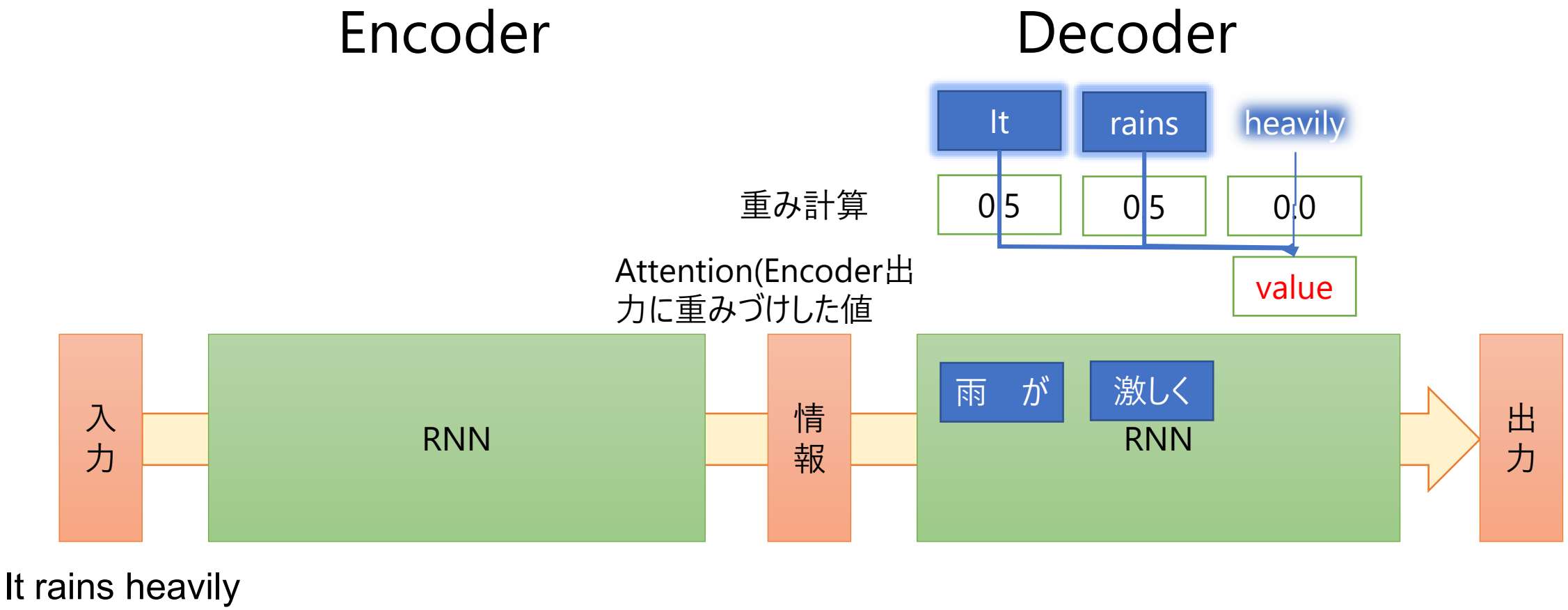
エンコーダーの出力全体のうち、特定部分に注目する



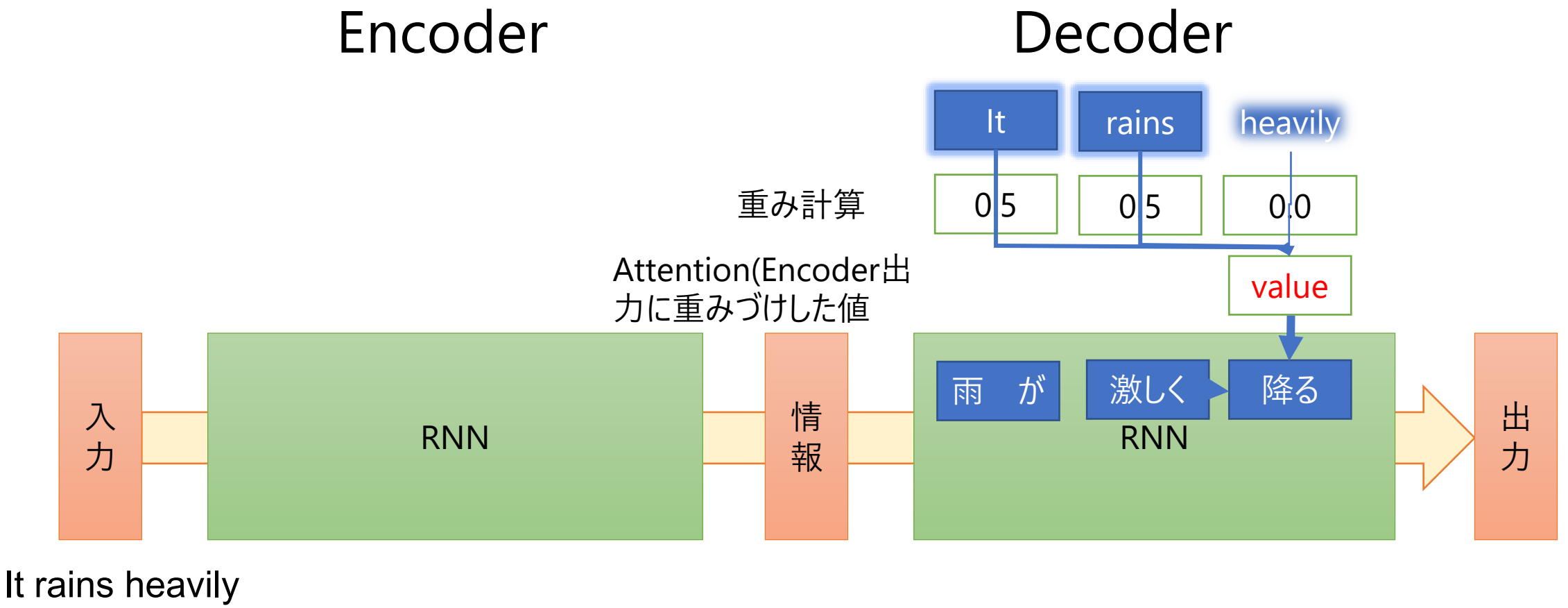
エンコーダーの出力全体のうち、特定部分に注目する



エンコーダーの出力全体のうち、特定部分に注目する



エンコーダーの出力全体のうち、特定部分に注目する



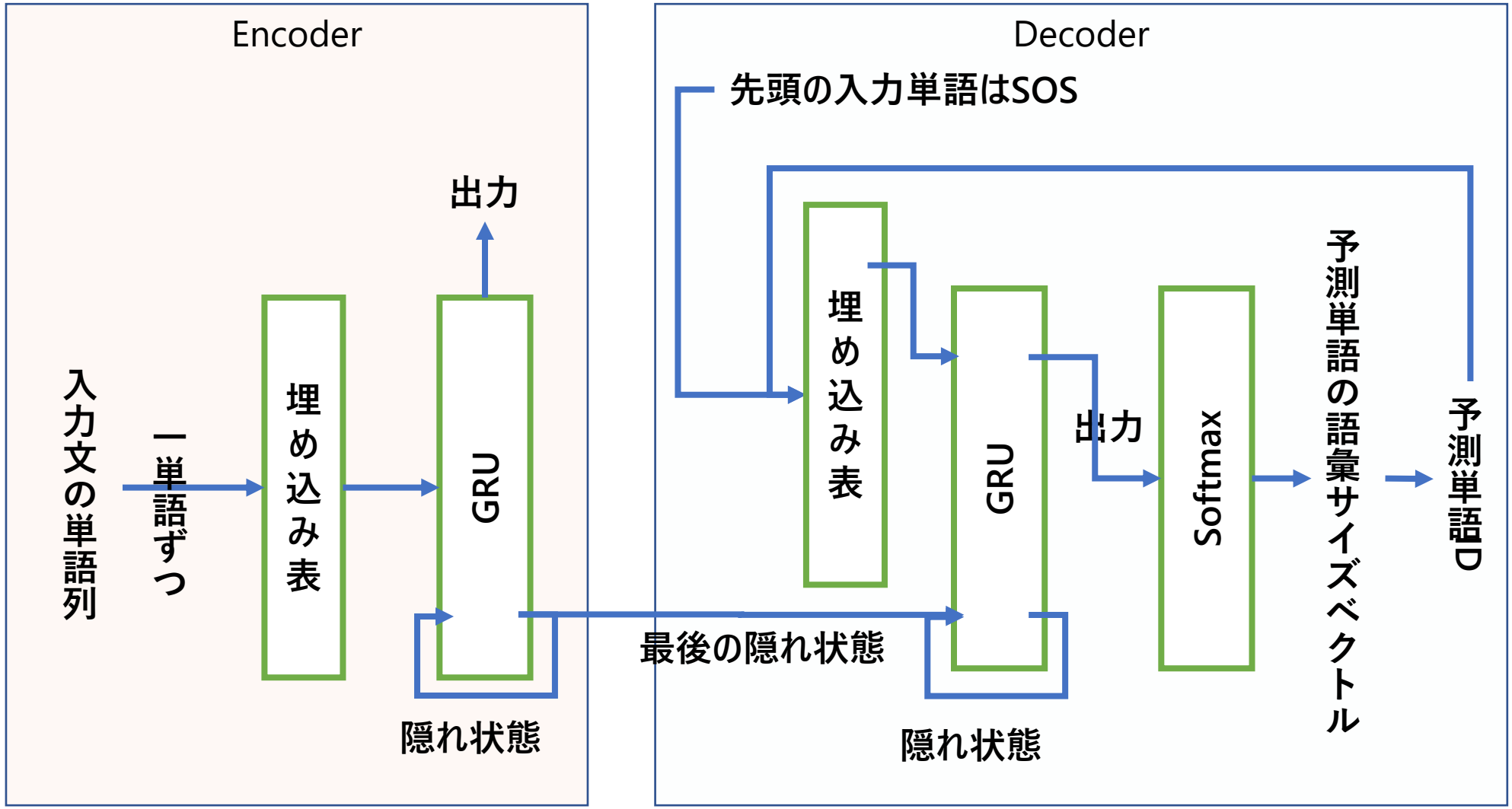
Attentionの学習

- Attentionは、ネットワーク構成中に、あるところに注目する仕組みを入れるというアプローチ。
- あるところに注目した結果、あるロスになった。それを踏まえて、逆伝播学習の中で、よりよい結果が得られるように注目個所を繰り返し最適化する。

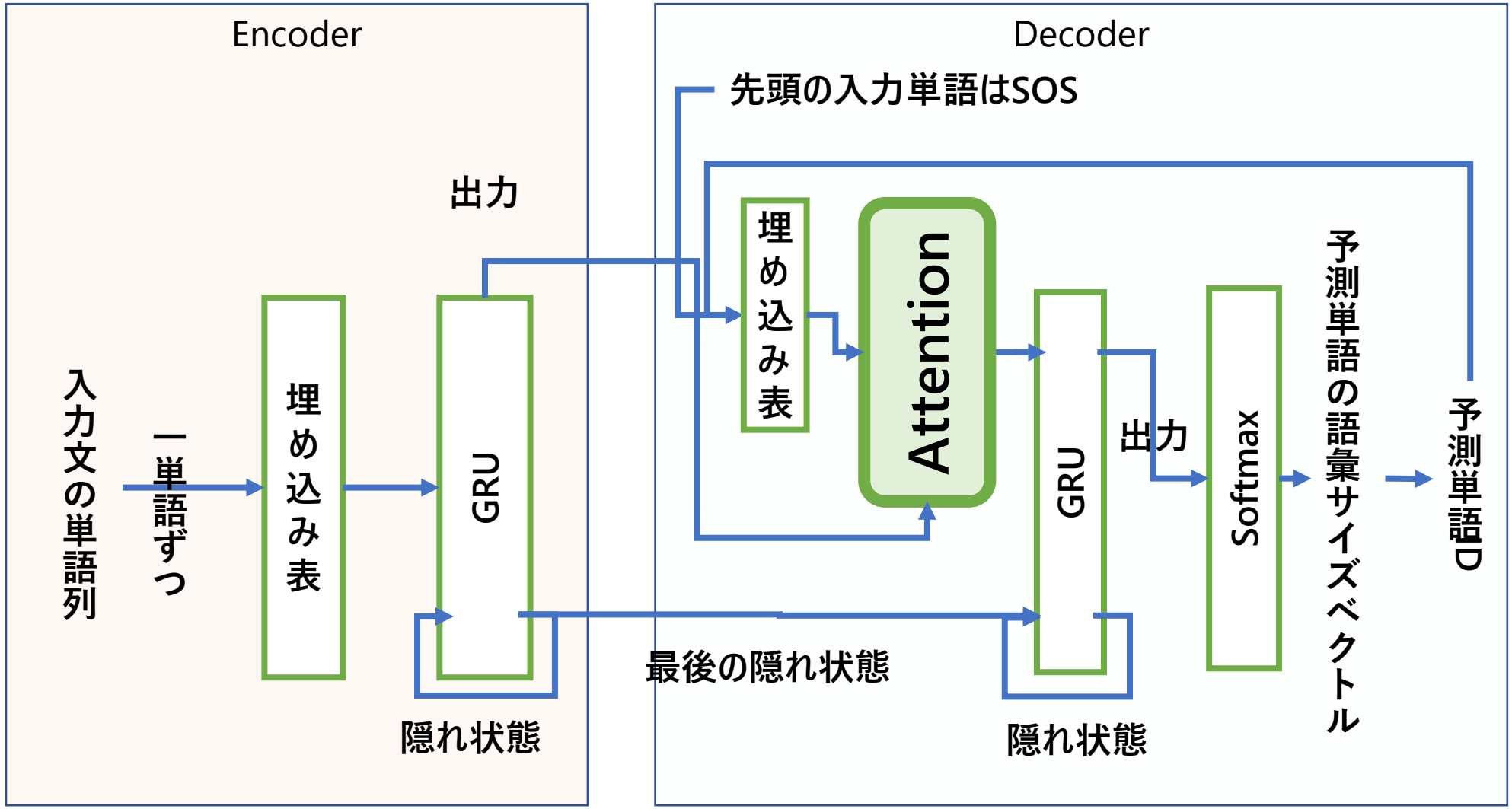
Attention 課題 1

- attention_translation.ipynb があります。rnn_translation の Attention込み版です。これを読解します。

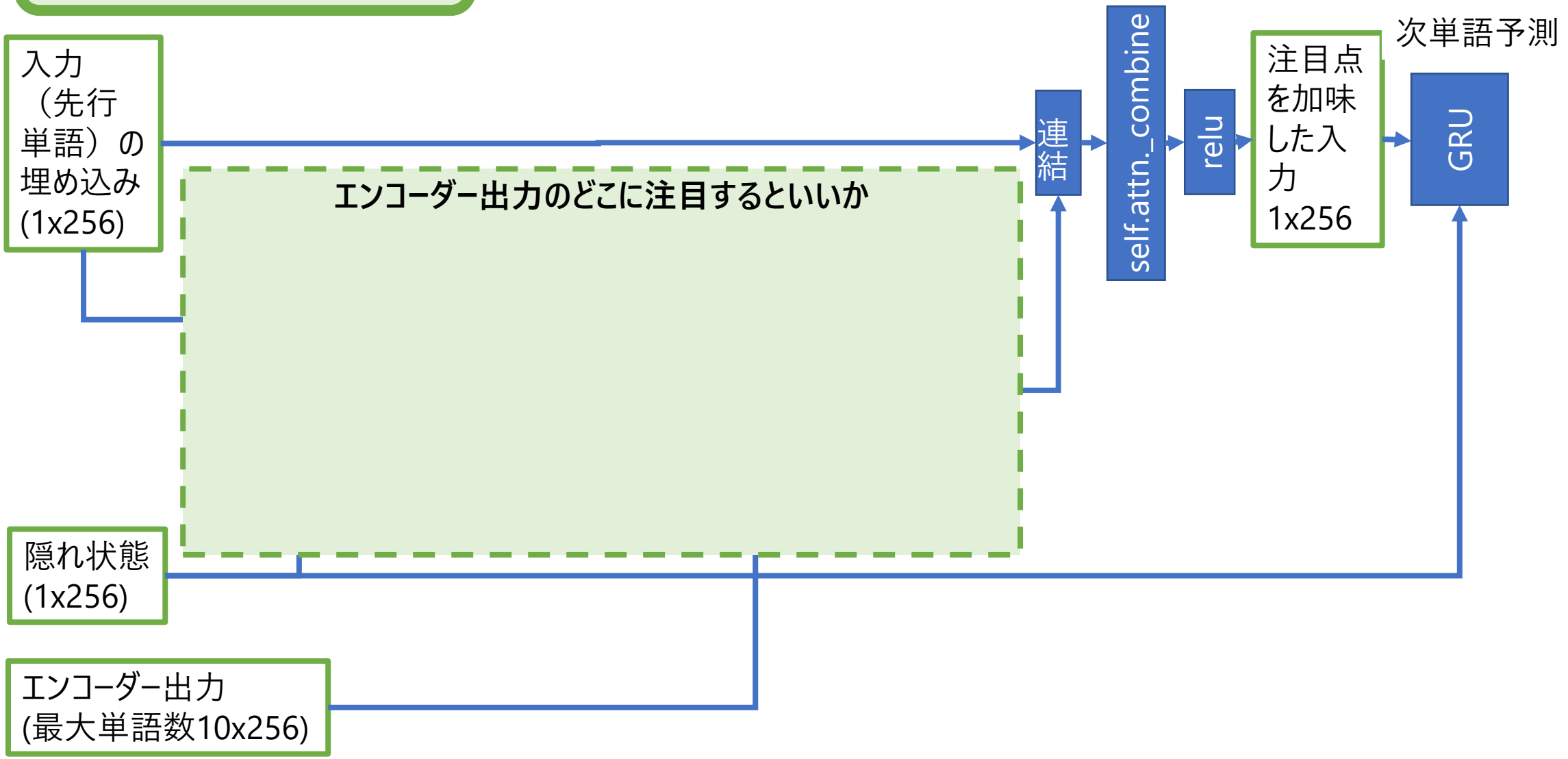
Attention入れる前のネットワーク構成



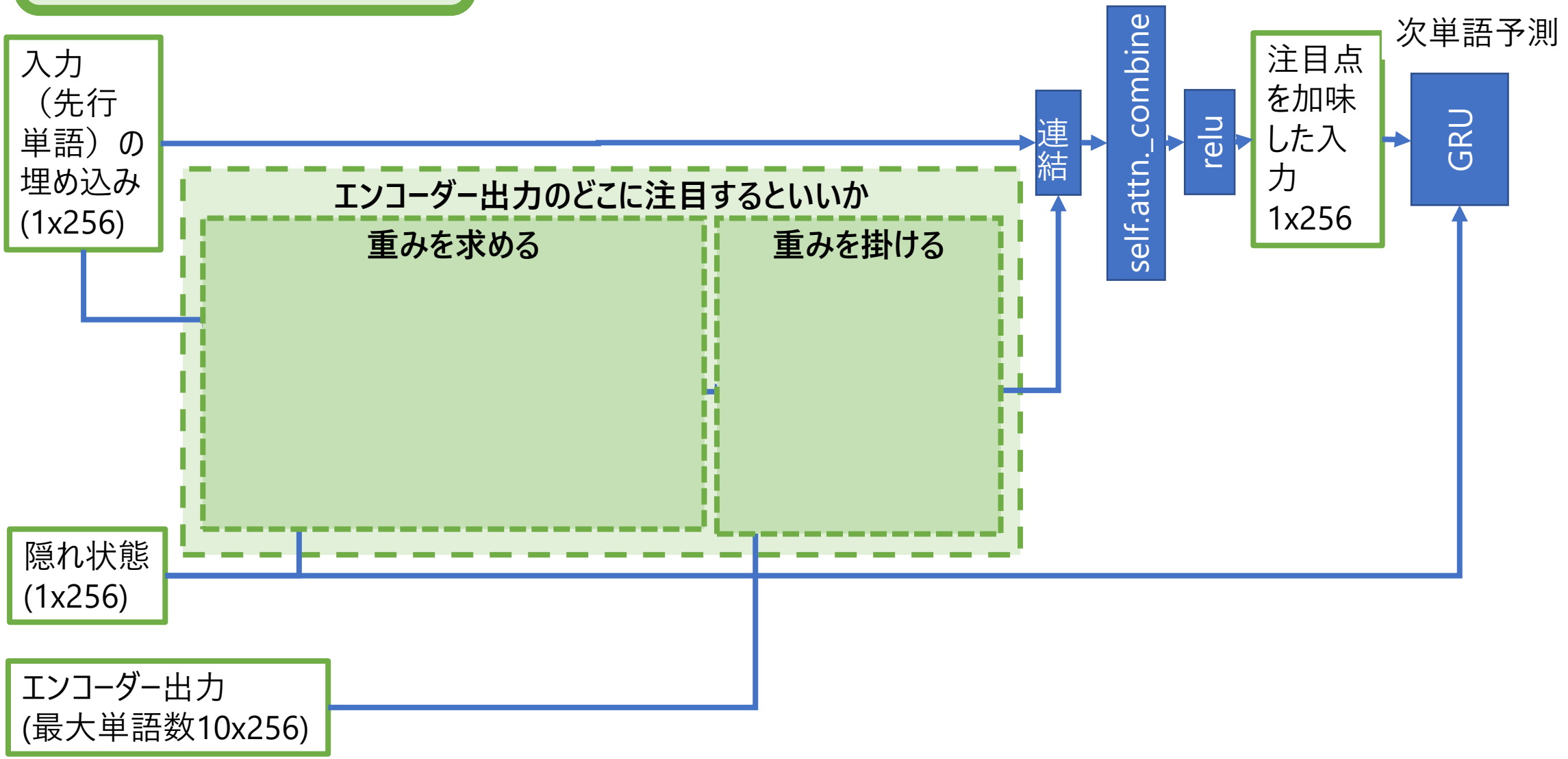
Attention組み込んだ後のネットワーク構成



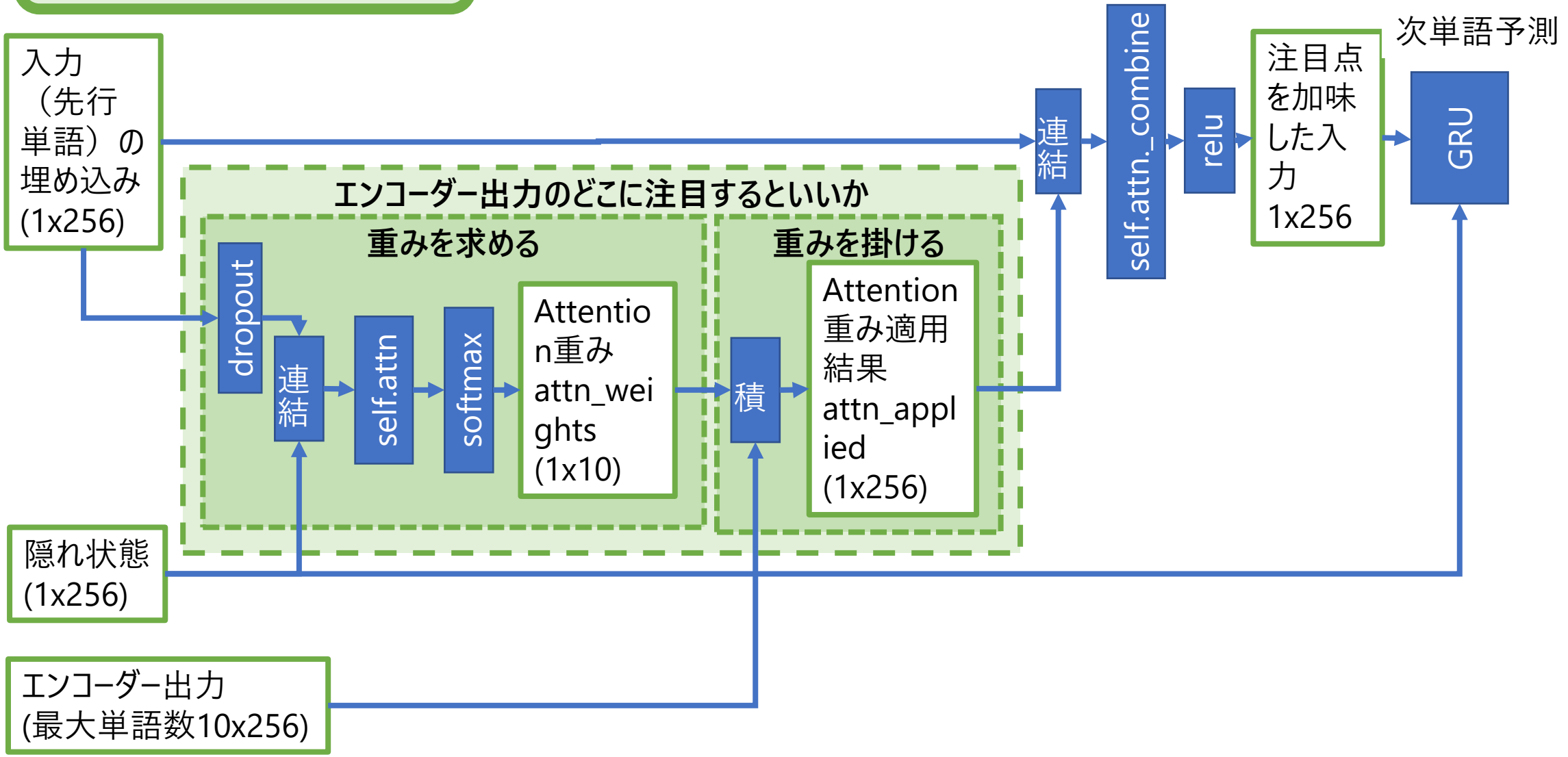
Attention


入力
（先行
単語）の
埋め込み
(1x256)
エンコーダー出力のどこに注目するといいか
連結
self.attn._combine
relu
注目点
を加味
した入
力
1x256
GRU
次単語予測
隠れ状態
(1x256)
エンコーダー出力
(最大単語数10x256)

Attention


入力
（先行
単語）の
埋め込み
(1x256)
エンコーダー出力のどこに注目するといいか
重みを求める
重みを掛ける
隠れ状態
(1x256)
エンコーダー出力
(最大単語数10x256)
連結
self.attn._combine
relu
注目点
を加味
した入
力
1x256
GRU
次単語予測

Attention


入力
（先行
単語）の
埋め込み
(1x256)
エンコーダー出力のどこに注目するといいか
重みを求める
dropout
連結
self.attn
softmax
Attentio
n重み
attn_wei
ghts
(1x10)
重みを掛ける
積
Attention
重み適用
結果
attn_appl
ied
(1x256)
連結
self.attn._combine
relu
注目点
を加味
した入
力
1x256
GRU
次単語予測
隠れ状態
(1x256)
エンコーダー出力
(最大単語数10x256)

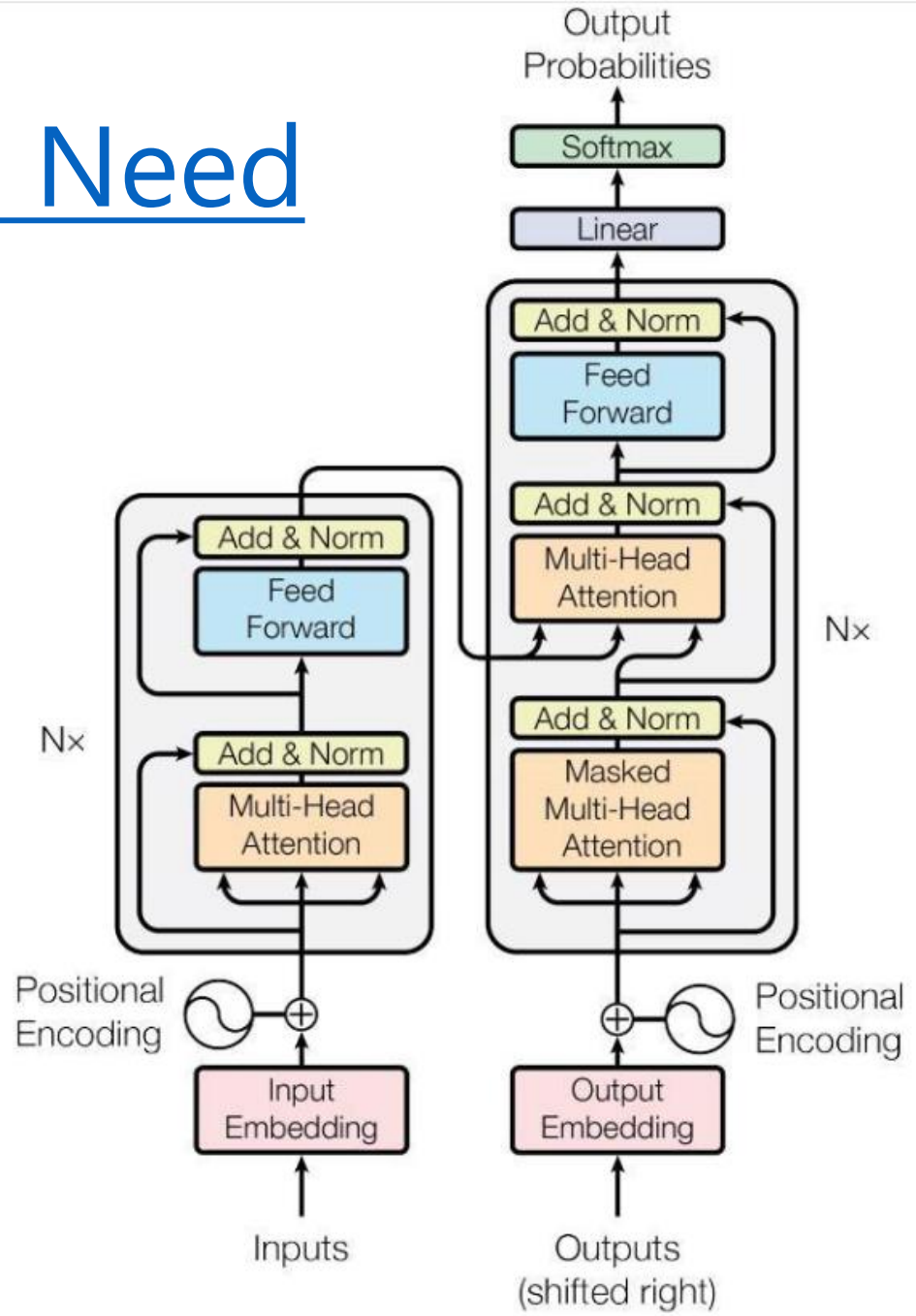
参考資料

- A Brief Overview of Attention Mechanism
- チュートリアル動画：Deep Learning入門：Attention（注意）、Sony 小林さん
- An introduction to Attention

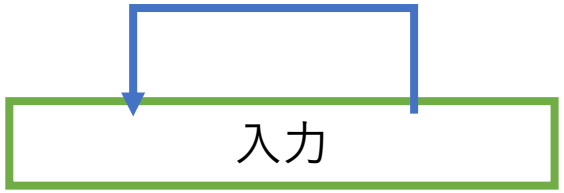
自然言語処理
Transformer

解説動画

Attention Is All You Need



Self-Attention

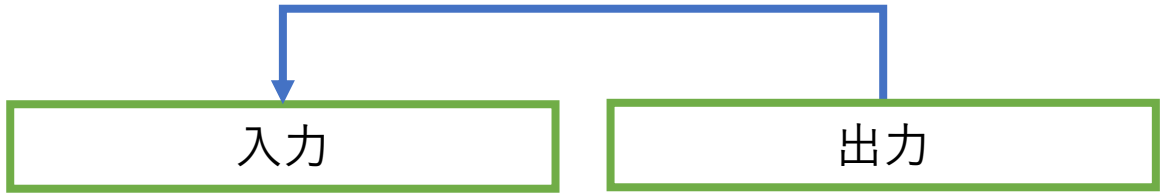


「鳥がナク」

「ナク」は鳥に注目すれば、
「泣く」でなくて「鳴く」。

ある入力文の中で、ある単語とほかの単語の関連度

Source-Target Attention



“It rains heavily”　　”雨が激しく降る”

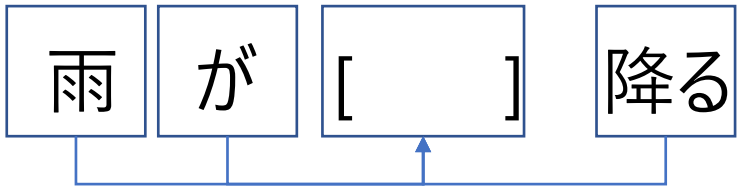
rainsに注目して、雨を選ぶ。

ある入力文と変換後の出力文の間で、ある出力単語と入力単語の関連度

Masked Language Model

雨　が　激しく　降る

雨　が　[　　]　降る　← ランダムに(複数単語を)マスク



← ほかの単語から穴を復元

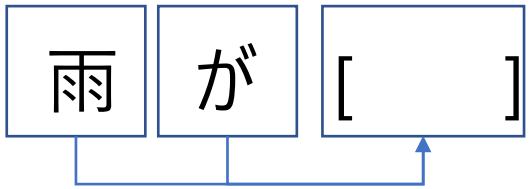
文章内の単語の相互の関連度を学習

Causal Language Model

雨　が　激しく　降る

雨　が　激しく　降る

先行単語を与えて、後続単語をマスク



先行単語から次単語を予測

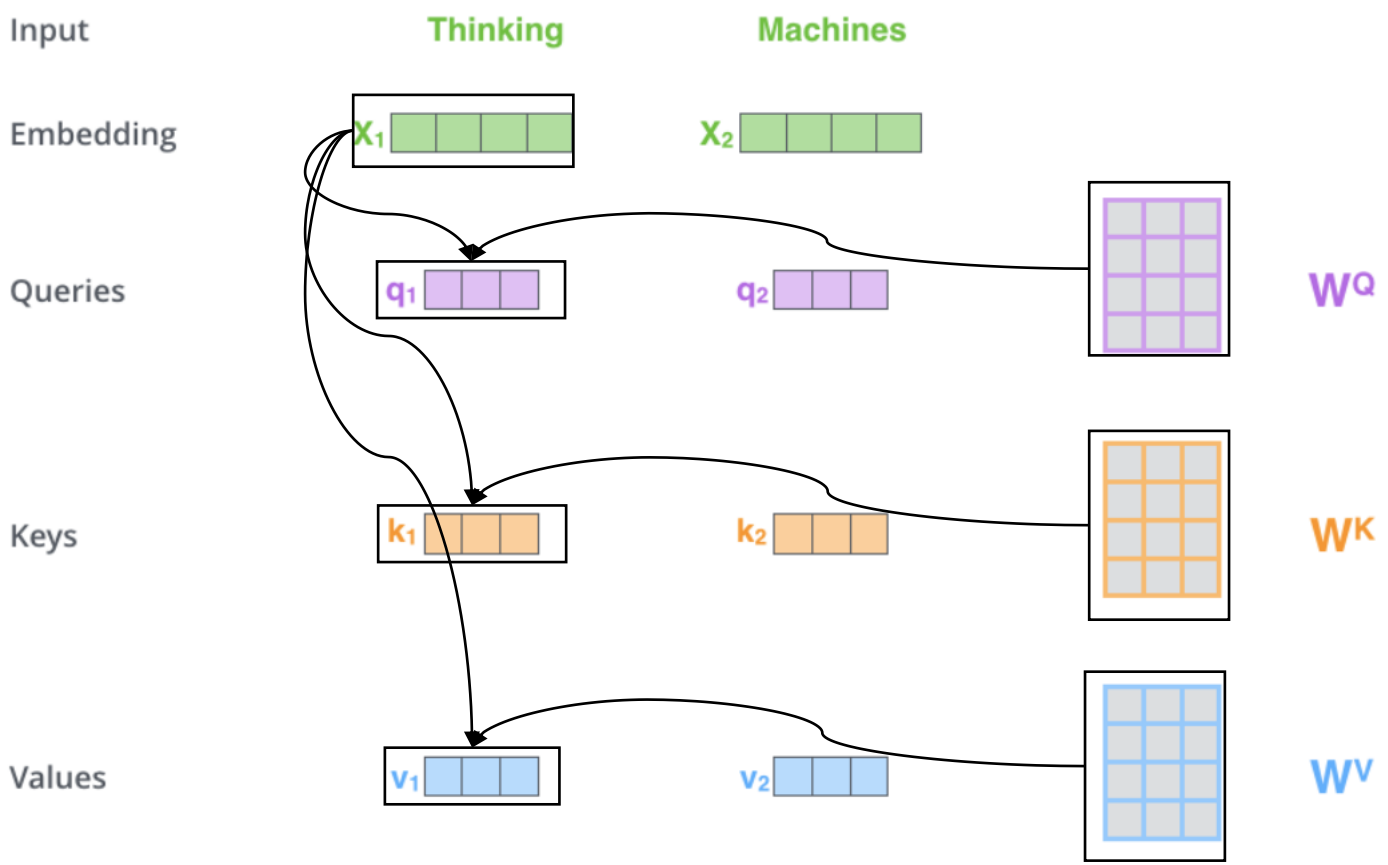
先行単語との関連度から次の単語を予測する

Transformer

huggingfaceの定義

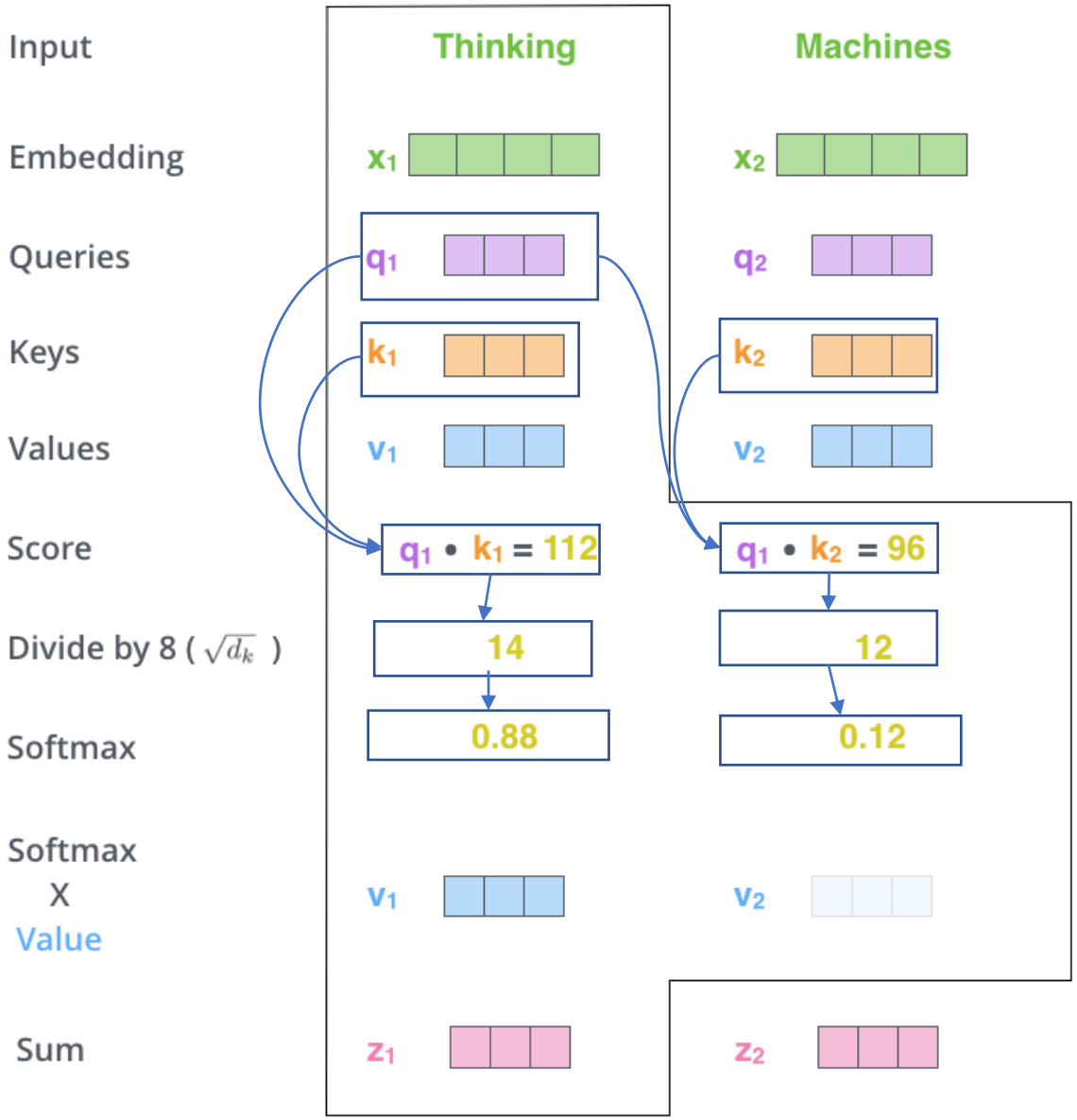
self-attention based deep learning model architecture

Query, Key, Value

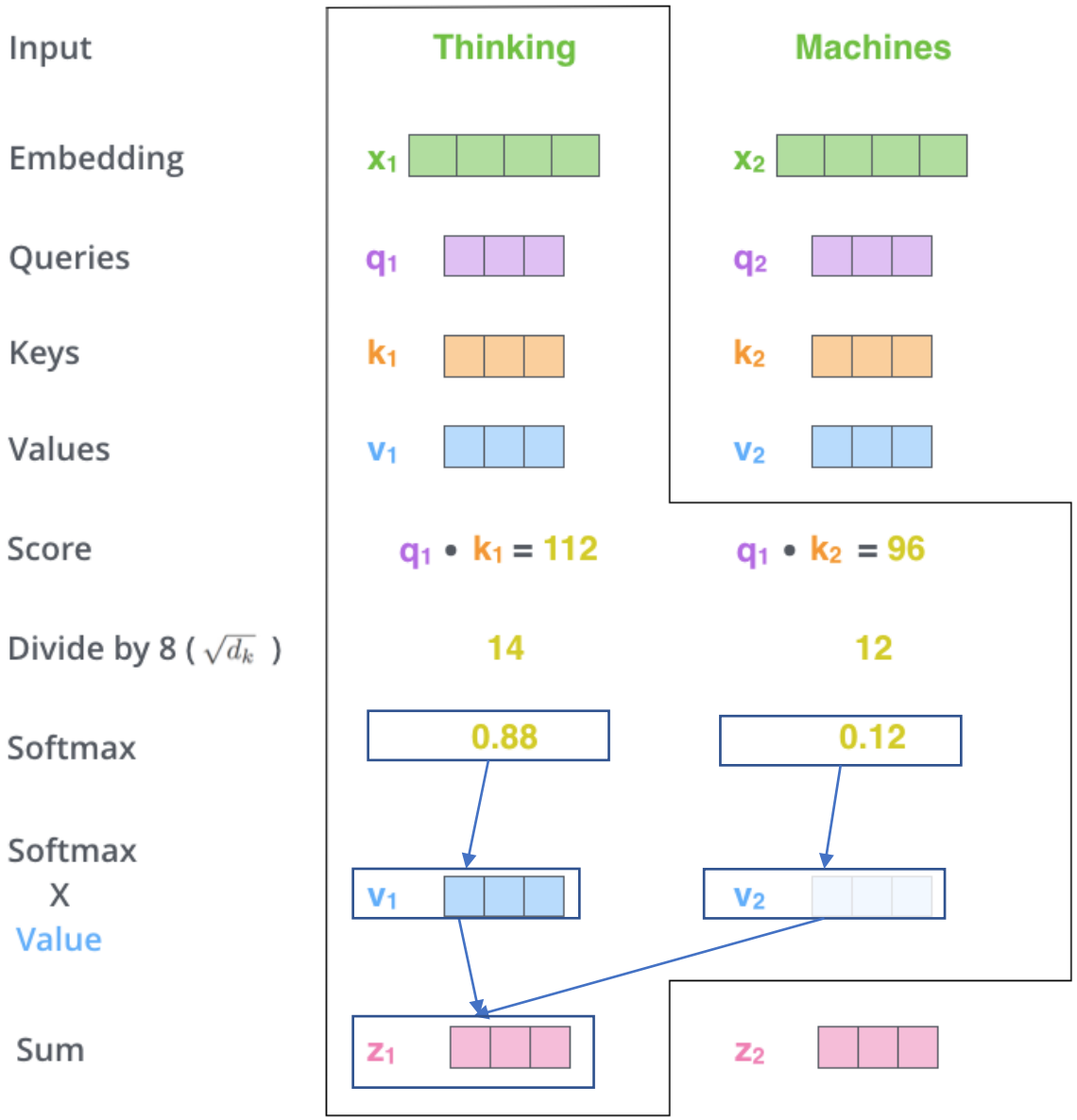


学習パラメータ

図はThe Illustrated Transformerから

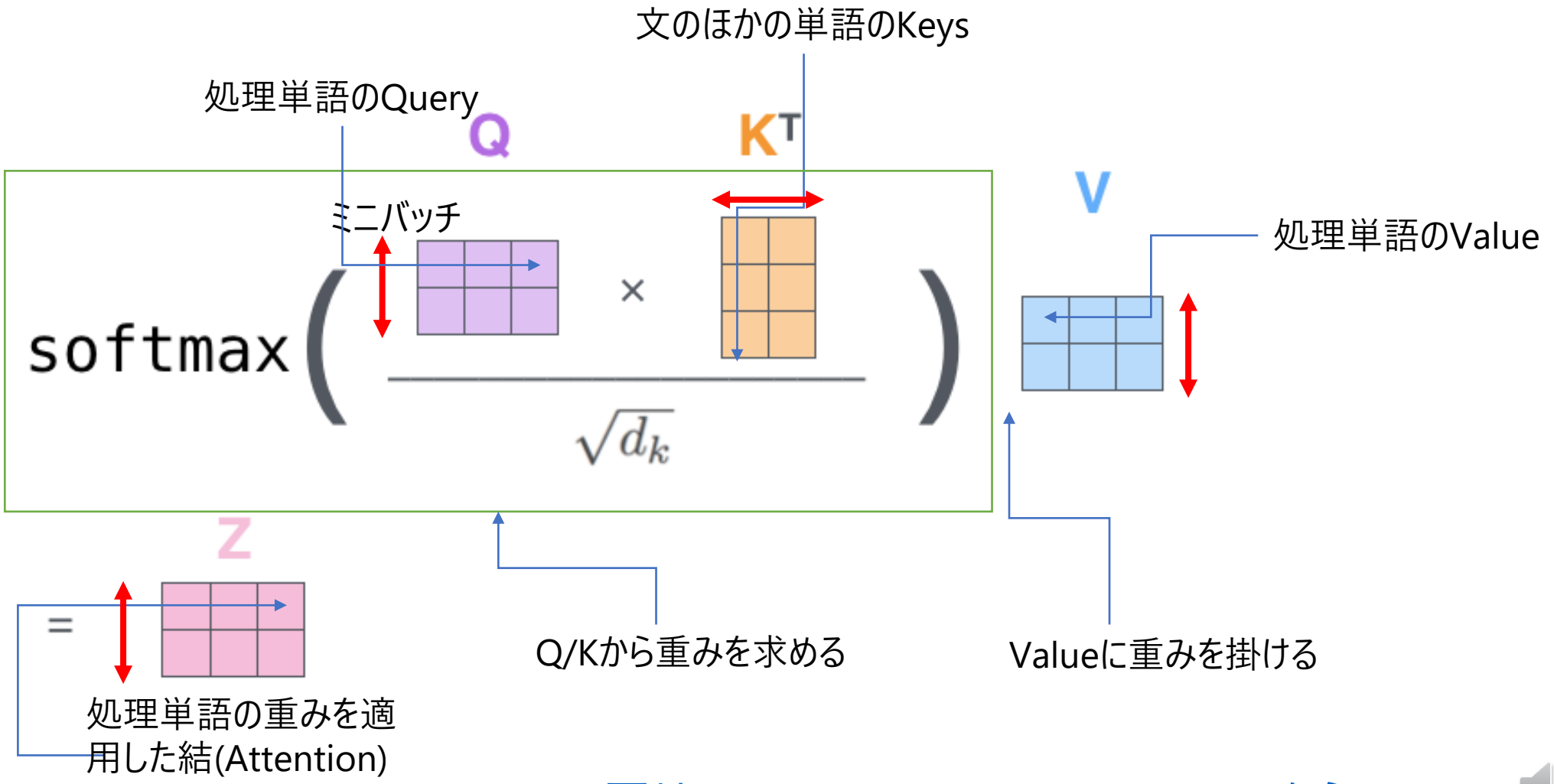


図はThe Illustrated Transformerから



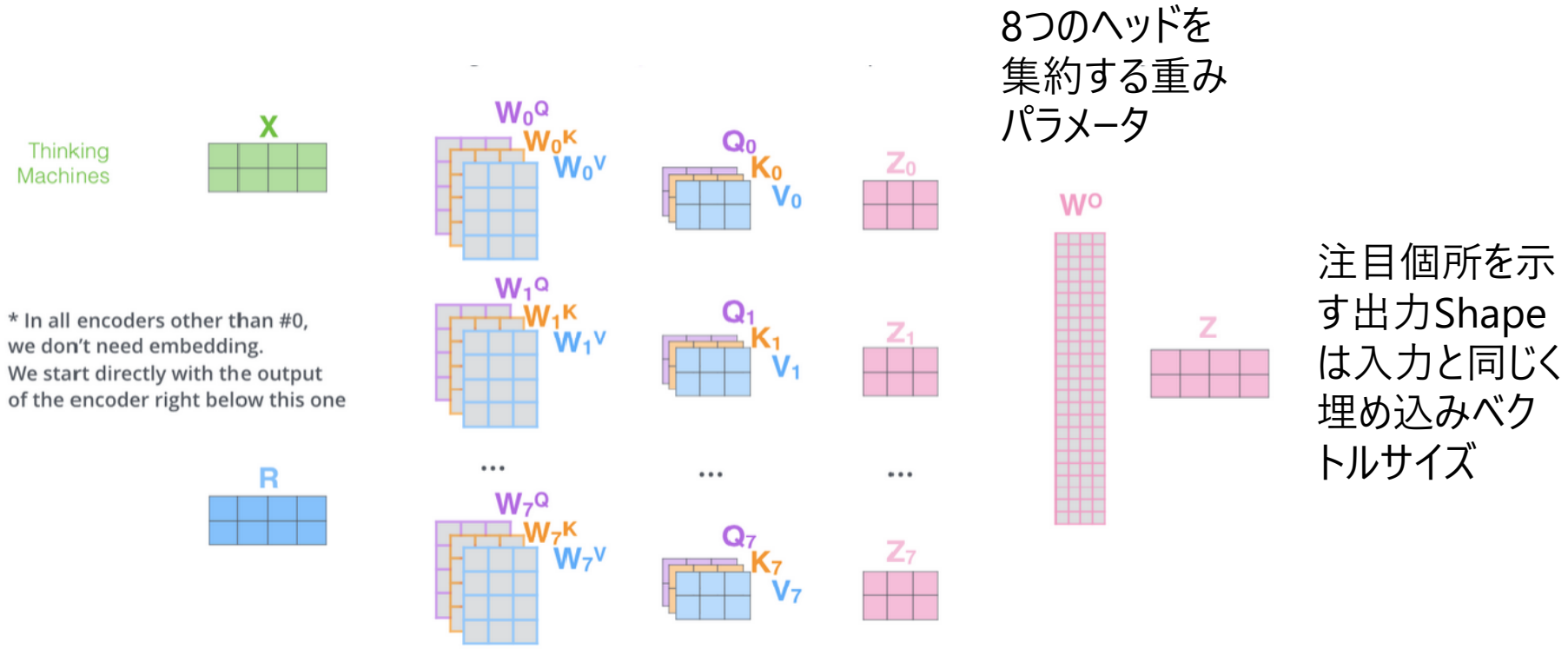
[図はThe Illustrated Transformerから]

Self-Attention



図はThe Illustrated Transformerから

マルチヘッドAttentionで複数の関連性を捕捉する



8つのヘッドを
集約する重み
パラメータ

注目個所を示
す出力Shape
は入力と同じく
埋め込みベク
トルサイズ

Q,K,Vを得る重
みパラメータ

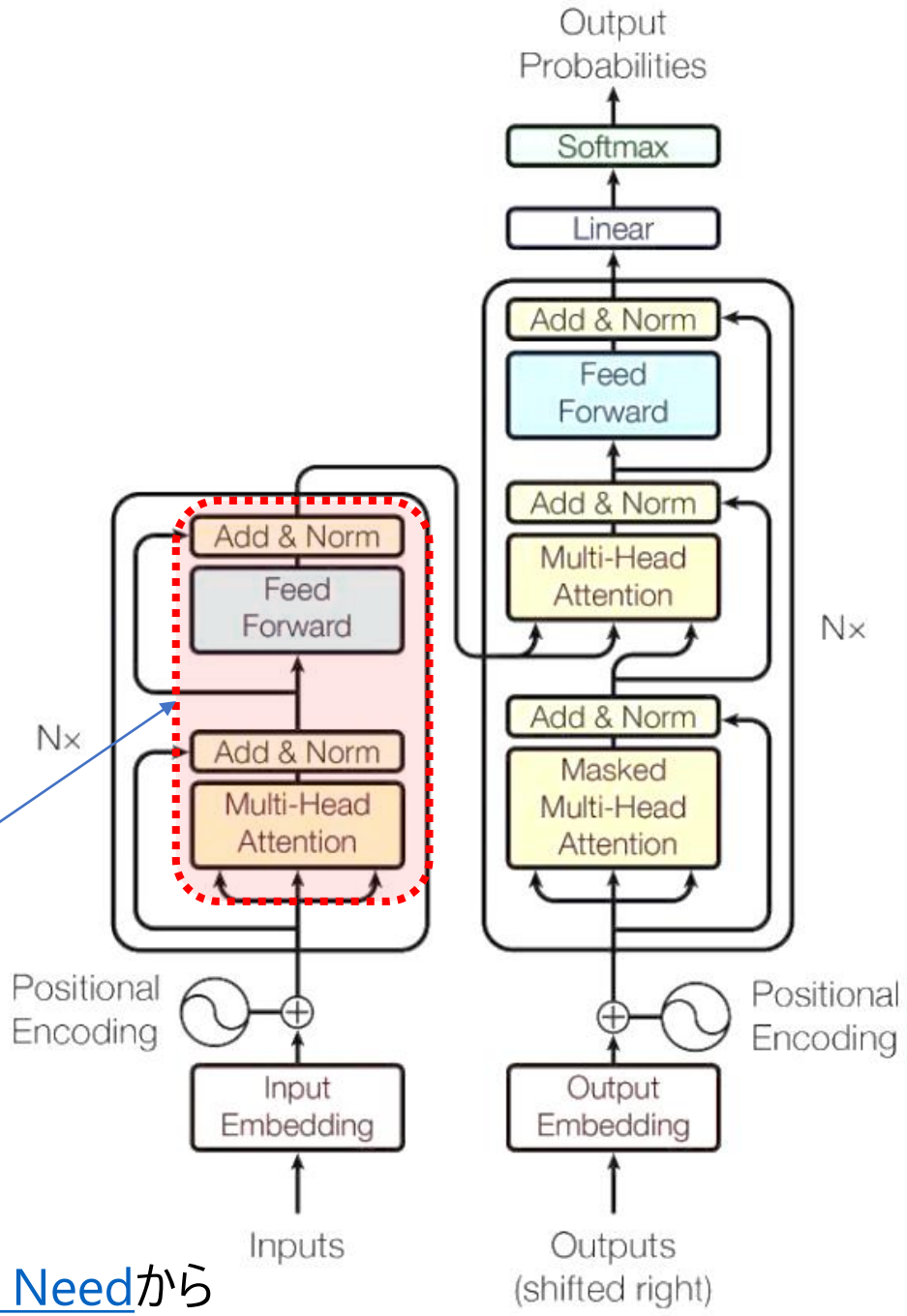
Q,K,V

Q,K,Vを適用した
注目個所を示す
Attention

[図はThe Illustrated Transformerから]

Self-AttentionをMasked LMでトレーニング。

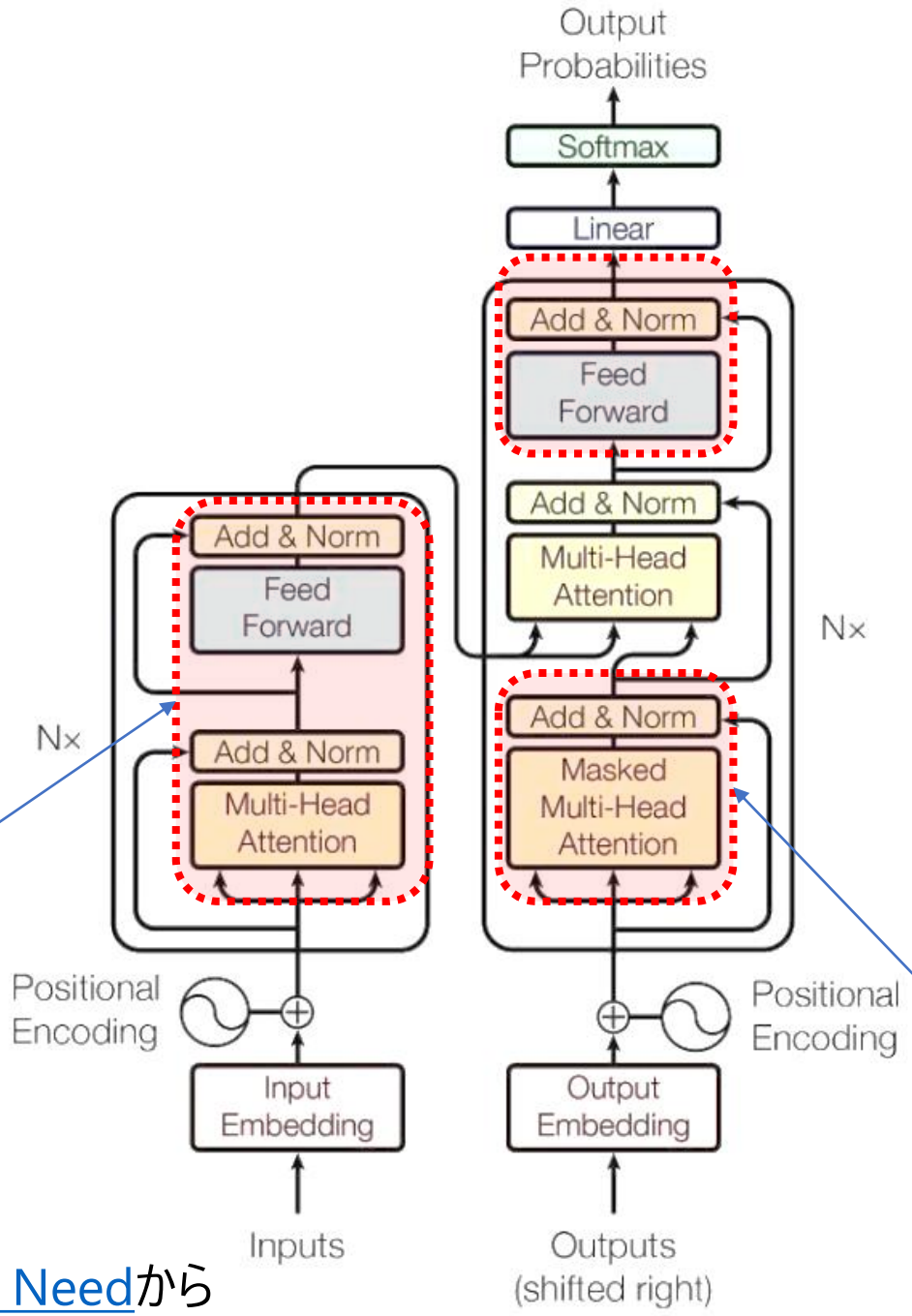
ソース言語内単語間関連性を学習。



図はAttention Is All You Needから

Self-Attentionを
Masked LMでトレーニング。

ソース言語内単語
間関連性を学習。



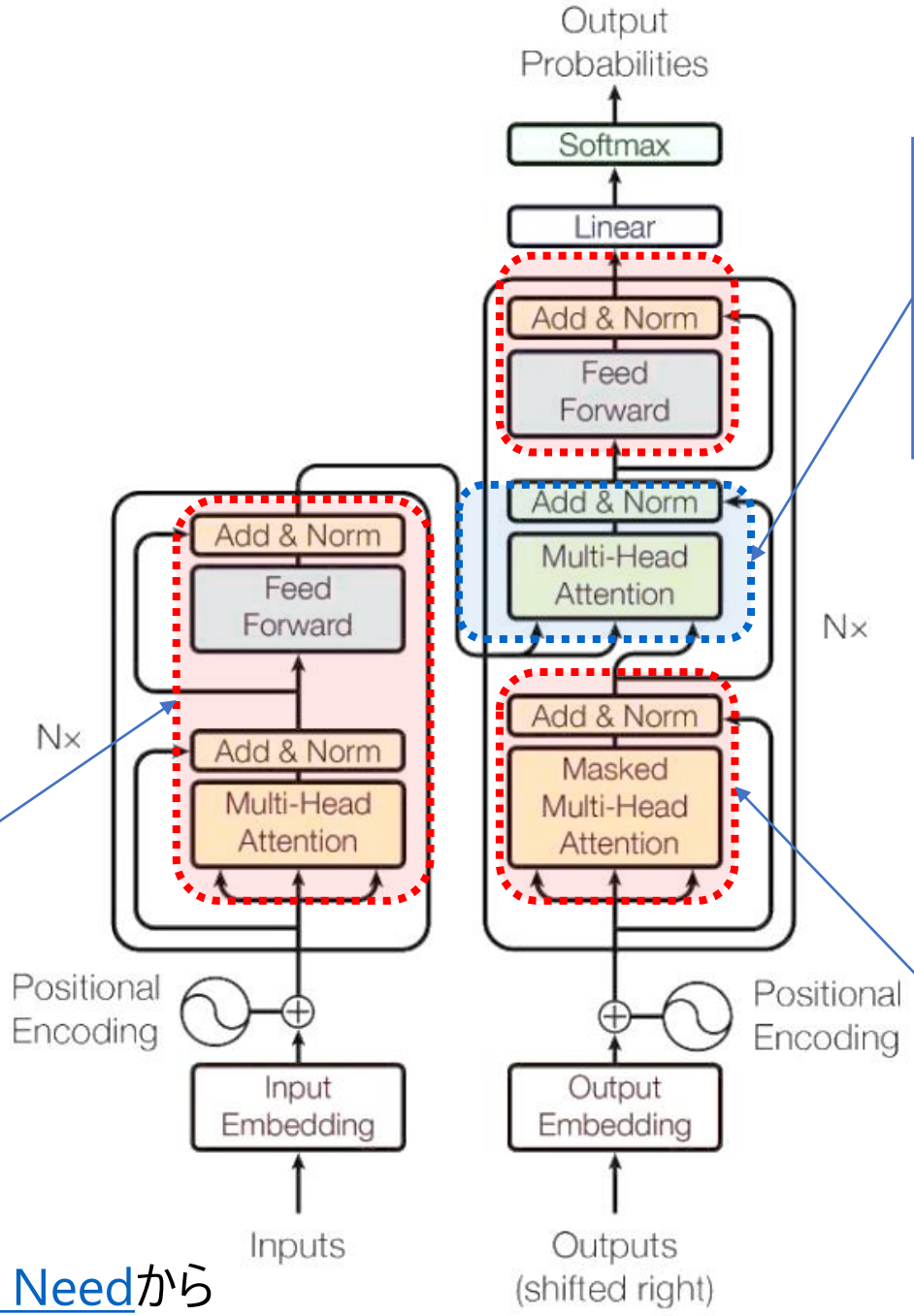
Self-Attentionを
Causal LMでトレーニング。

ターゲット言語内単語
間関連性から次単語
予測できるように学習。

図はAttention Is All You Needから

Self-Attentionを
Masked LMでトレーニング。

ソース言語内単語間関連性を学習。



Source-Target Attention。

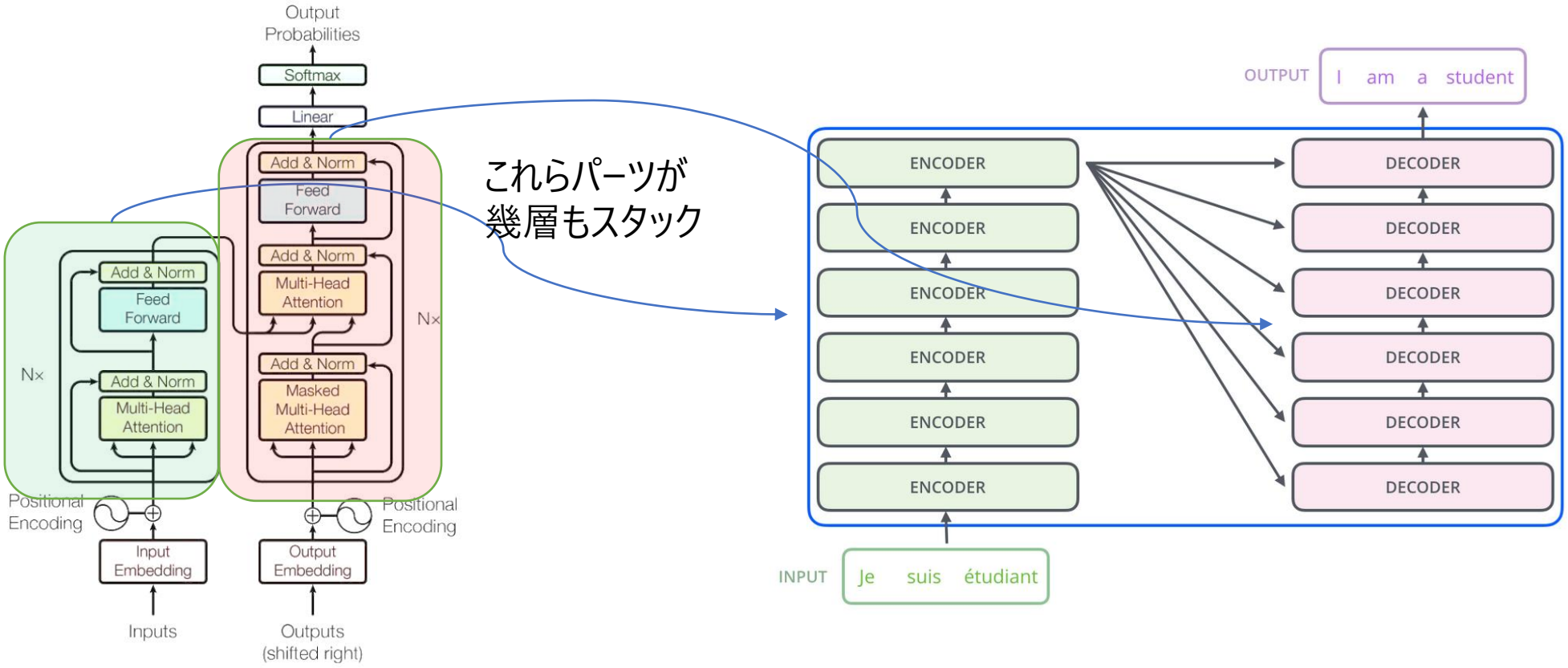
ソース文とターゲット文間単語関連性を利用。

Self-Attentionを
Causal LMでトレーニング。

ターゲット言語内単語間関連性から次単語予測できるように学習。

図はAttention Is All You Needから

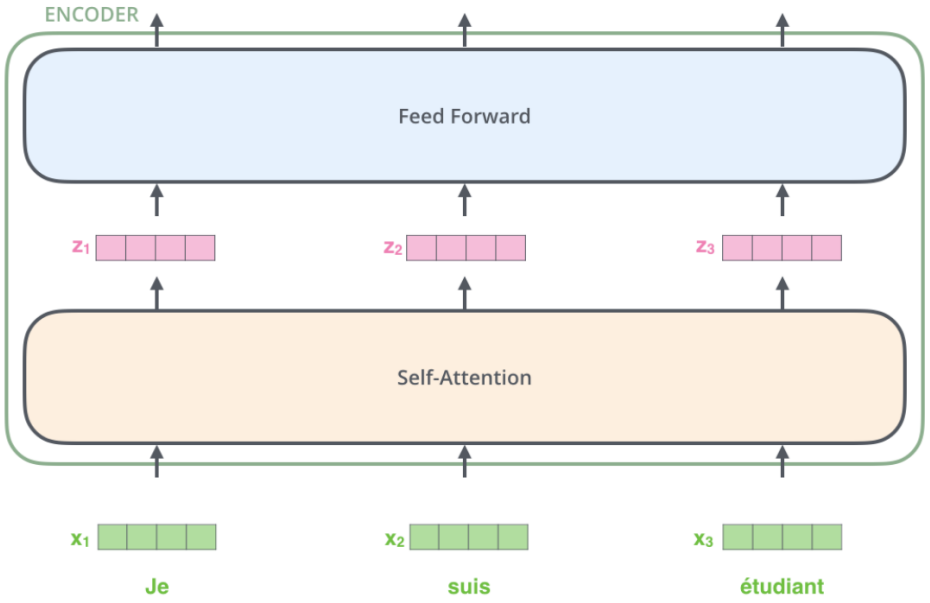
BERT全体構成：レイヤのスタック



[図はAttention Is All You Need から]

[図はThe Illustrated Transformer から]

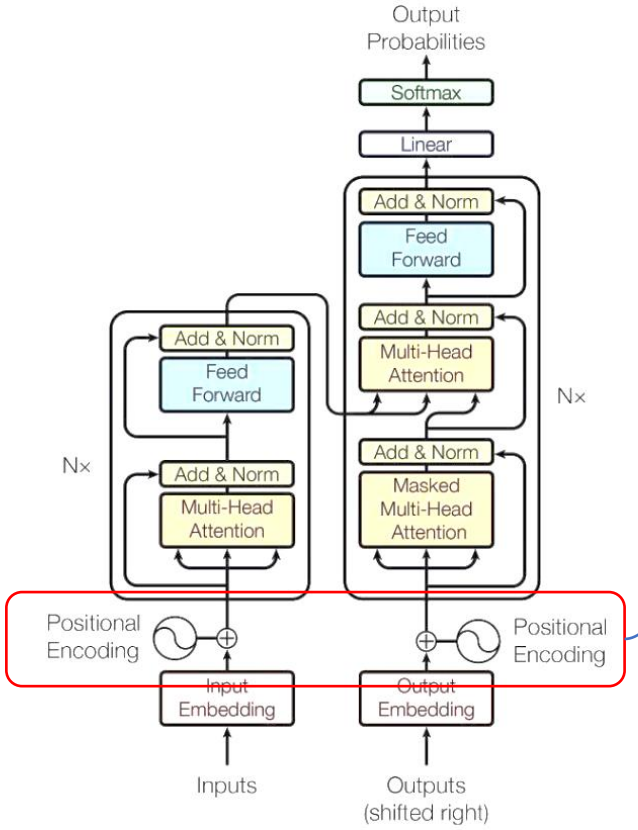
単語の埋め込みベクトルが、順次上のレイヤに

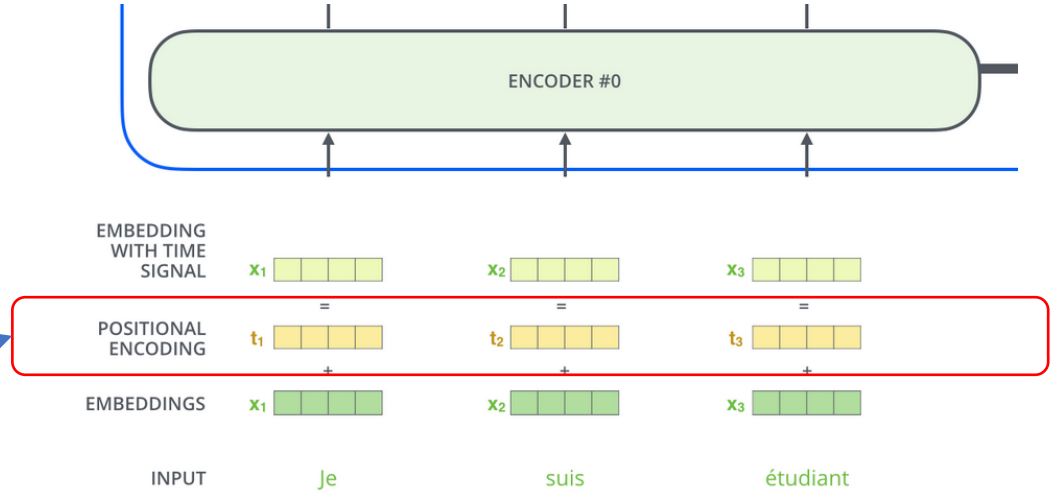


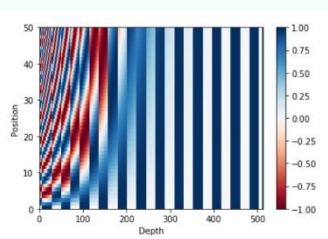
単語ごとに全結合

図はThe Illustrated Transformer から

Positional Encoding：系列情報を埋め込みベクトルに加える





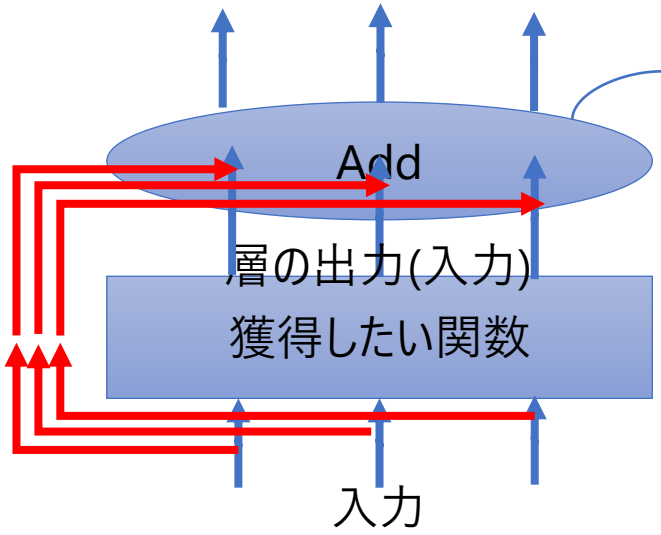


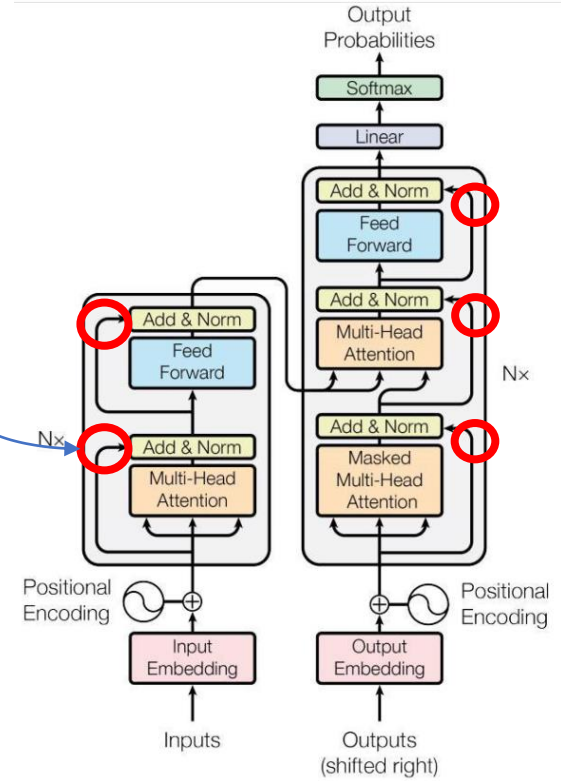
図はAttention Is All You Need から

図はThe Illustrated Transformer から

Residual Connection

層の出力(入力) = 獲得したい関数(入力) – 入力
⇒
獲得したい関数(入力) = 層の出力(入力) + 入力



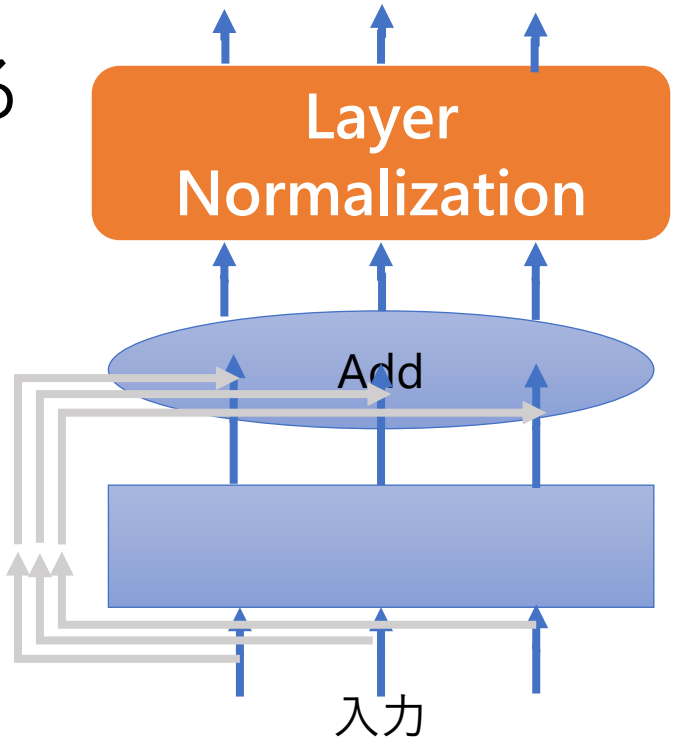


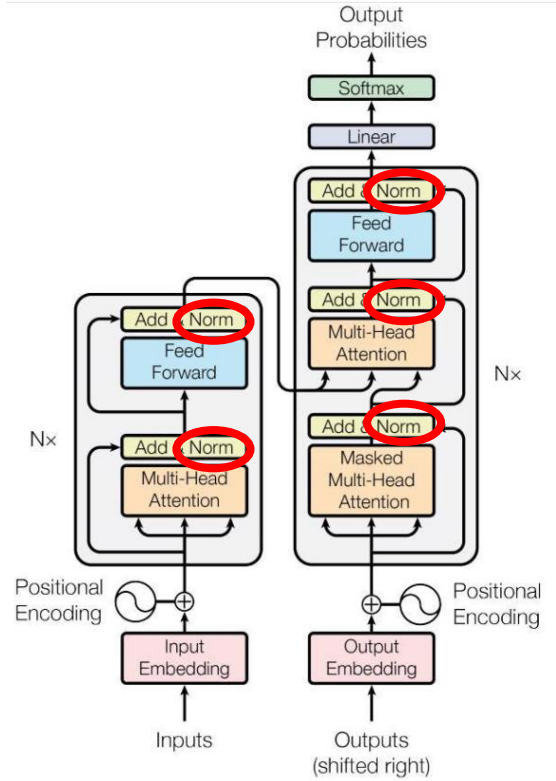
層を単純に重ねると学習しにくくなる。一方、層を深くして複雑な構造を学習させたい。
⇒ Residual Connectionで層を深くしても学習するようにする

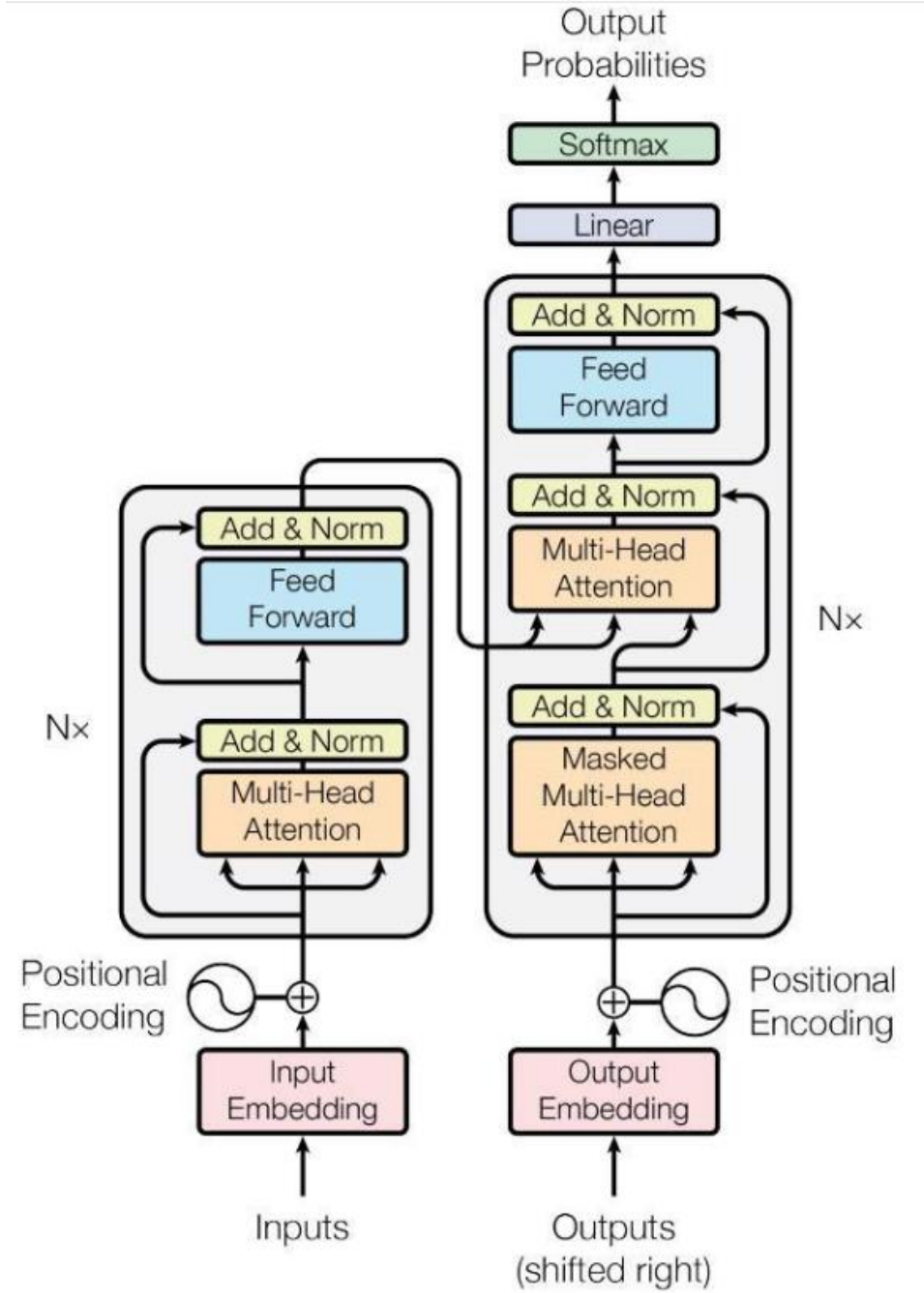
Layer Normalization

平均と分散を見て、ばらつきを抑える

⇒学習の高速化、過学習抑制






Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

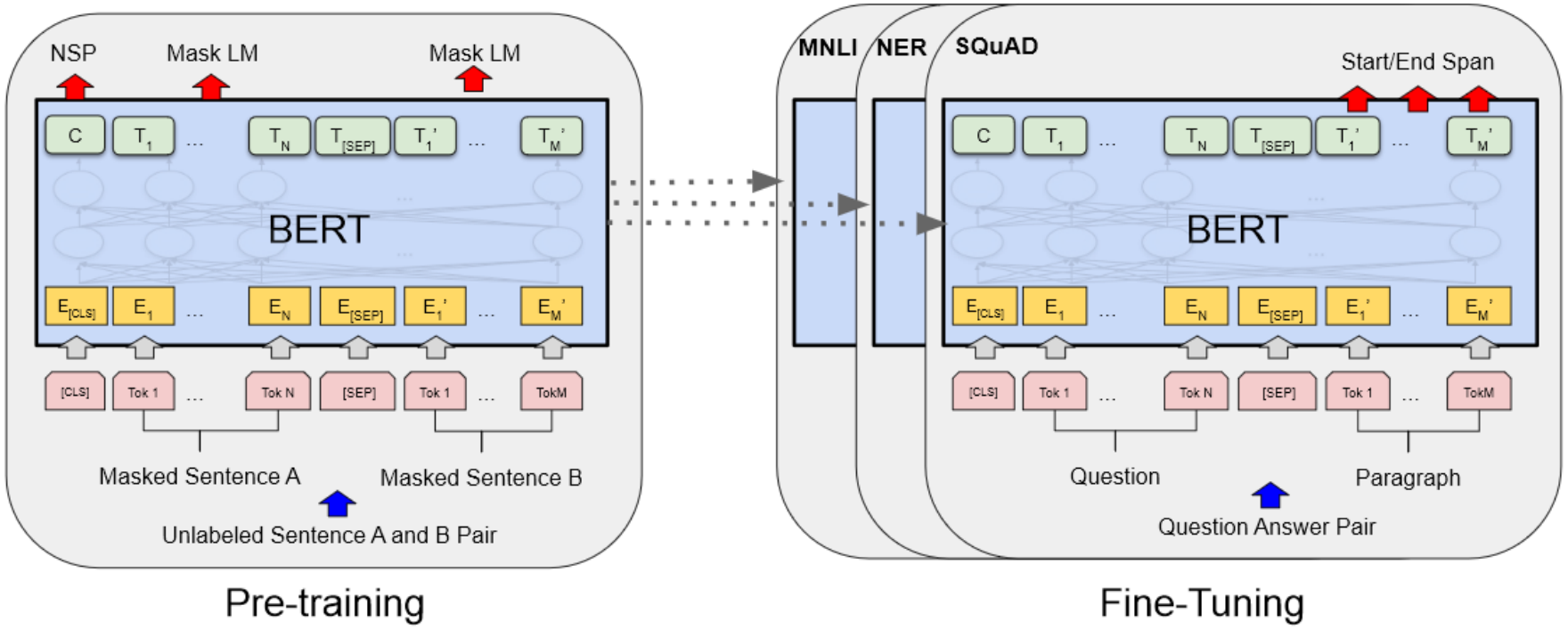
効果：SoTA(State-of-The-Art)記録を塗り替えた

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

[Attention Is All You Need, 2017]

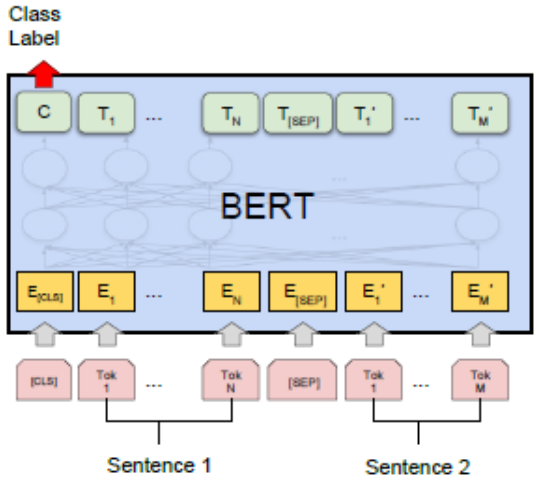
Fine-Tuning：事前にトレーニングしたBERTの出力を、タスクに応じたデータで再度Tuning



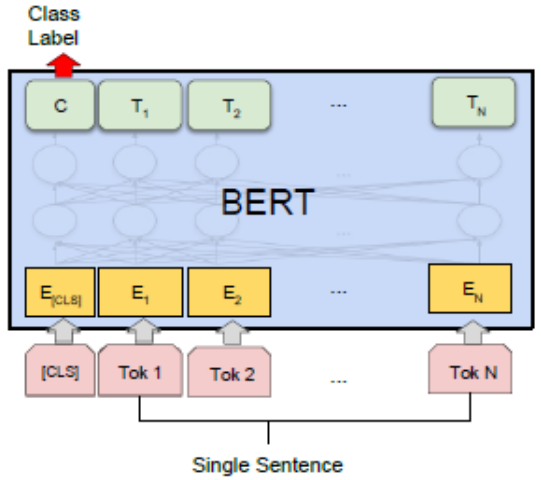
[BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding、2018]

取り出す情報を変えて、タスクに合わせる

出力の先頭ラベルで相次ぐ文の関係の分類



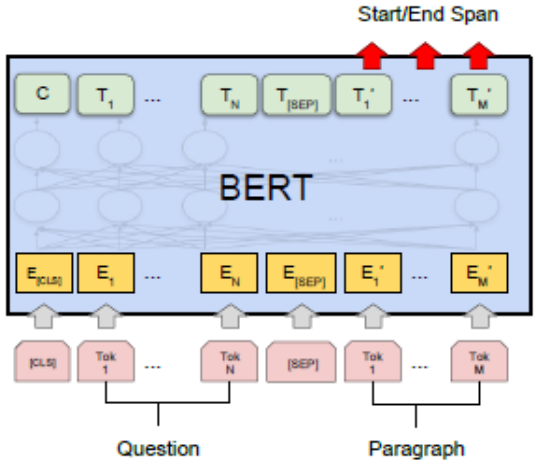
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



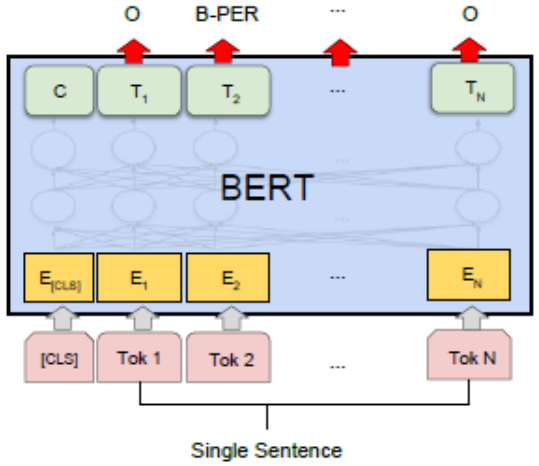
(b) Single Sentence Classification Tasks: SST-2, CoLA

出力の先頭ラベルで文の分類

次文の開始マーカ・終了マーカを取り出して、次文予測



(c) Question Answering Tasks: SQuAD v1.1



(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

単語にタグ付け（場所表現取り出しとか)

[BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding、2018]

HuggingfaceのTransformerの分類

自動回帰モデル

- Causal LM
- 応用：文生成
- 例：GPT、BERTのデコーダー

自動符号化モデル

- Masked LM
- 応用：文分類、単語タギング
- 例：BERTのエンコーダー

SEQ2SEQモデル

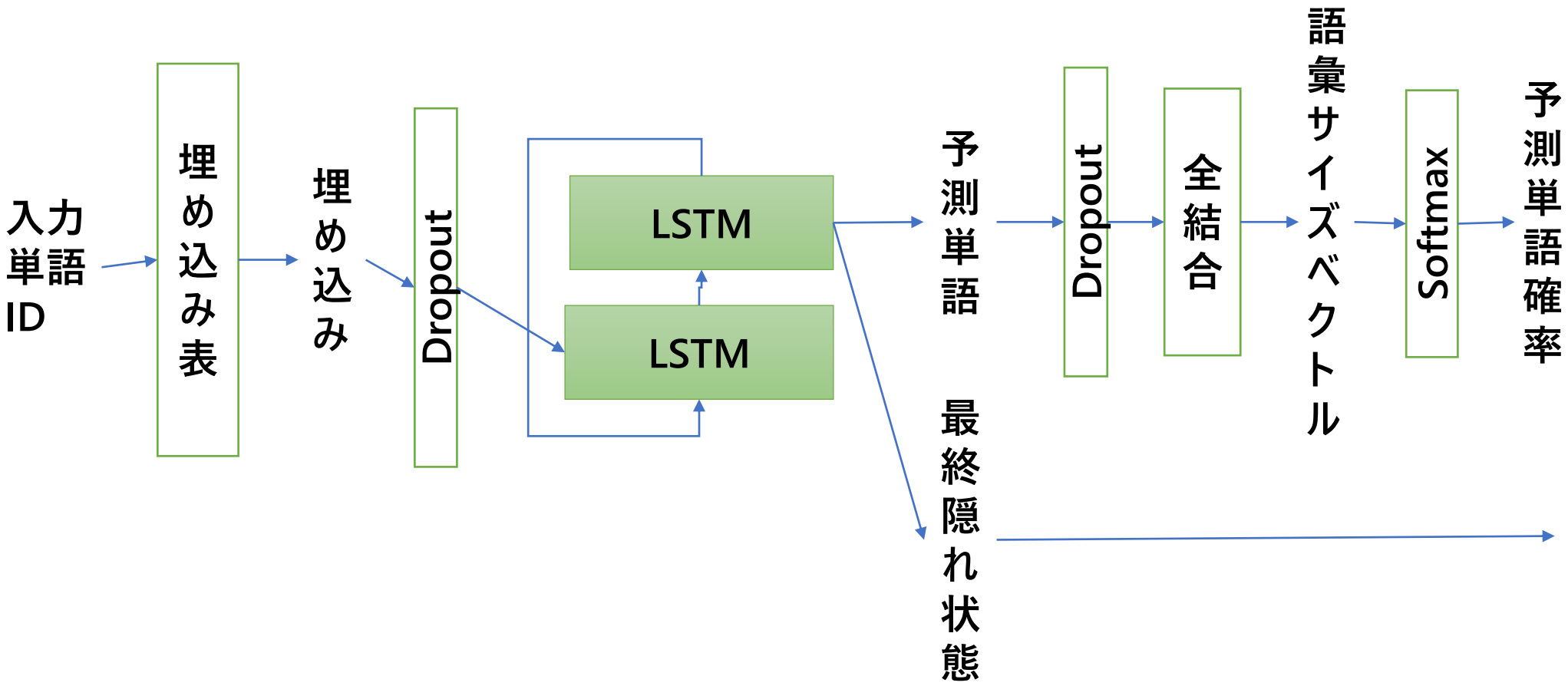
- エンコーダー・デコーダー構成のもの
- 応用：機械翻訳、要約、Q&A
- 例：BERTの翻訳タスク版、T5

https://huggingface.co/transformers/model_summary.html

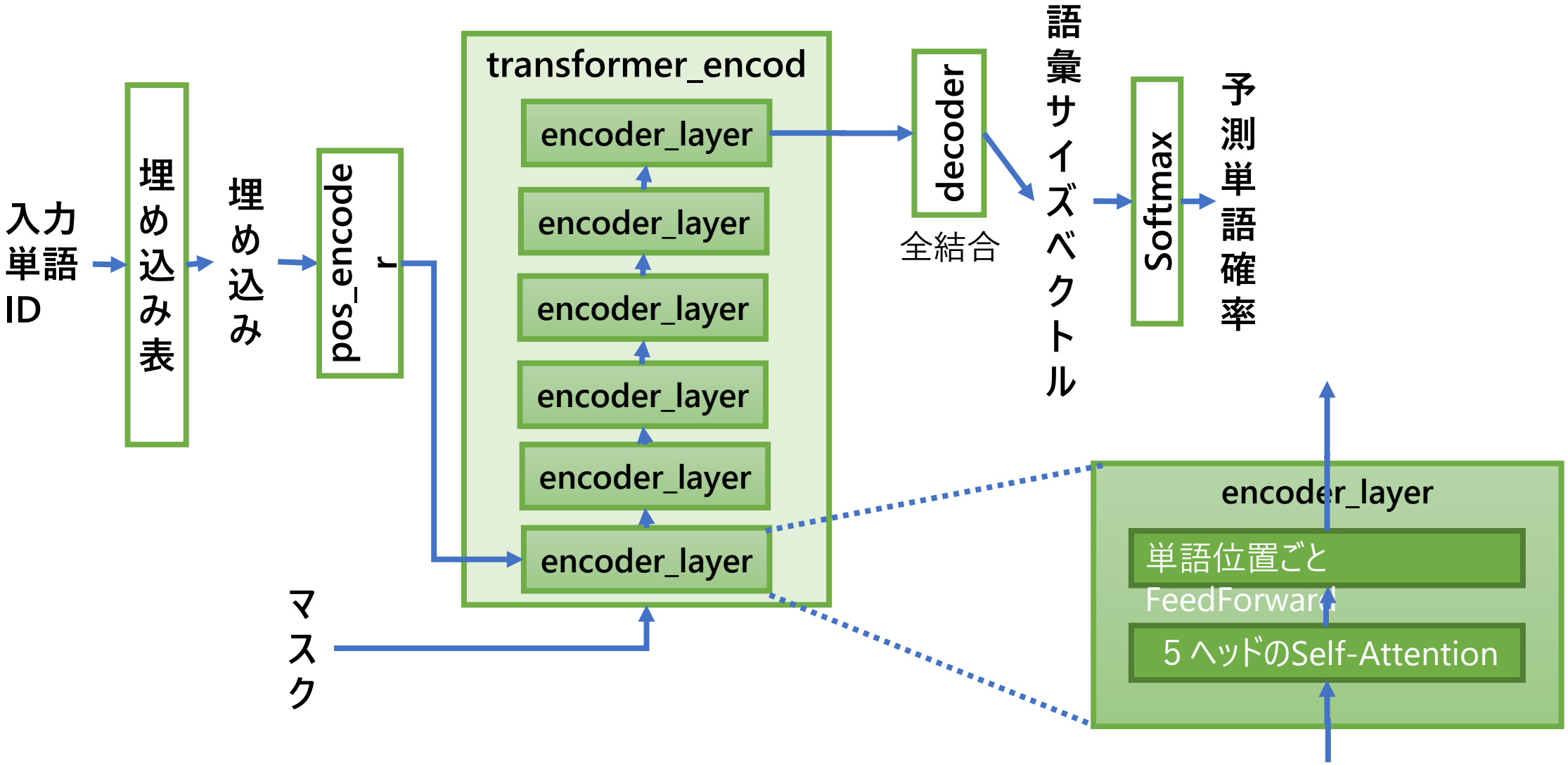
Transformer 課題 1

- attention_languagemodel.ipynb があります。rnn_languagemodel のサンプルの Transformer 版です。これを読解します。
- 実行ログを追加て、提出しなさい。

振り返り：LSTMを使った言語モデルのネット構成



Transformerを使った言語モデルのネット構成



参考資料：Transformer,BERT

- Original Paper
  - Attention Is All You Need, 201
  - BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018
- The Illustrated Transformer
- 作って理解する Transformer / Attention
- Harvard NLP, The Annotated Transformer
- 深層学習界の大前提Transformerの論文解説！

参考資料：画像処理でのAttention

- Self-Attentionを全面的に使った新時代の画像認識モデルを解説！
- Exploring Self-attention for Image Recognition

100本ノック第9章課題89

- ニューラルネットのコードに慣れてきました。「100本ノック」の9章の課題 の89(Transformer)の回答例を読みましょう。
- 「CNNRNNTransformer.ipynb」というノートをコピーし、89の回答例コードを読解してください。80をやってから実行ログを残してください。

RNN,CNN,Transformerを利用するコードが理解できたら、すでに、高度なテーマでも、NLPとDeep Learningを自力で深めていく力がついています。

確認クイズ

- スタログの確認クイズをやってください。

86（ミニバッチサイズ＝１）