BERT

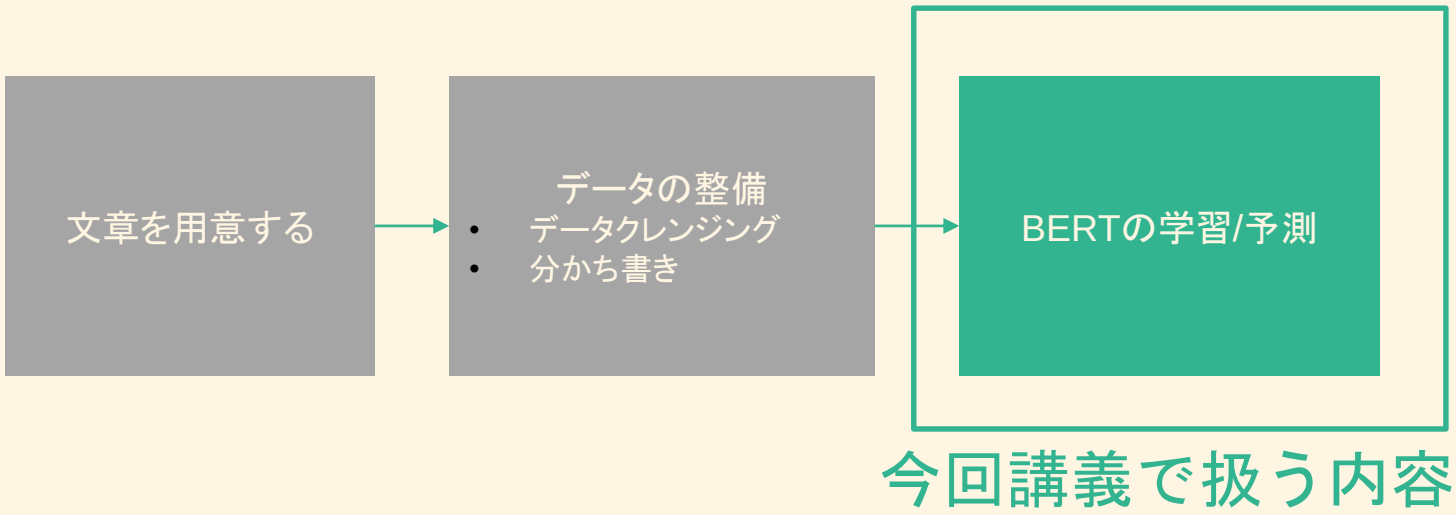
Nature Insight
ネイチャーインサイト 株式会社

目次

この講義の目的

講義の目的

今回の講義では、BERTの理論・実装・応用例について解説していきます。
データの整備は取り扱いませんが、この部分は既知として進めますので、
もし不安な方は自然言語処理の基礎講義の受講で復習してからこの講義を受けてください。



1. BERTの概要

BERTの概要

Nature Insight

ニューラルネットワークを用いた自然言語処理の歴史

分類	モデル	時期	説明
単語の分散表現	word2vec	2003年~	単語の分散表現を単純なニューラルネットワークで表現する手法。「王様」-「男」+「女」=「女王」と計算できることがとても衝撃的だった。
RNNベース	RNN	2010年？~	前の単語の情報を考慮して機械学習を行う手法。 当時は順番を考慮する手法がほとんどなかったため、重宝された。
	LSTM/GRU	2014年	RNNの進化版。RNNの欠点である「ほとんど直前の情報しか使われず、何個か前の情報はほとんど使われない」を修正したもの。 現在でも時系列予測や文章生成などは、LSTMやGRUベースのモデルが使用されている。
	LSTM+Attention	2015年~2017年	LSTMの機構に「Attention」という相互に単語を考慮する機構を入れることで、精度向上を図ったもの。 このAttentionがのちのBERTに大きな影響を与えた。
Attentionベース	Transformer	2017年	LSTMを用いずに、Attentionのみでネットワークを構成したモデル。TransformerはEncoder-Decoderモデルで、翻訳や文章生成で使用された。
	BERT	2018年~	TransformerのEncoder部分を取り出して、事前学習させたもの。 BERTで学習させたモデルは当時多くのタスクでSotAを更新し、話題を呼んだ。

BERTの概要

Nature Insight

BERTの特徴

- BERTは様々なタスクでSotAを更新したモデルである。
- BERTは事前学習済みモデルであり、汎用的なモデルである。
- BERTはWikipedia等の巨大なコーパスから教師なし学習で作成される。
- BERTは文脈を双方向に学習するモデルである。
- BERTは周辺文脈を考慮して単語の特徴量を決定できる。

BERTの概要

Nature Insight

BERTは様々なタスクでSotAを更新したモデルである。

BERTは様々なタスクでBERTは当時のモデルに対し、様々なベンチマークタスクでSotA(State of the Art)を更新したモデルです。

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	Average -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

評価サーバー（https://gluebenchmark.com/leaderboard）によって採点されたGLUEテスト結果。

BERTの概要

Nature Insight

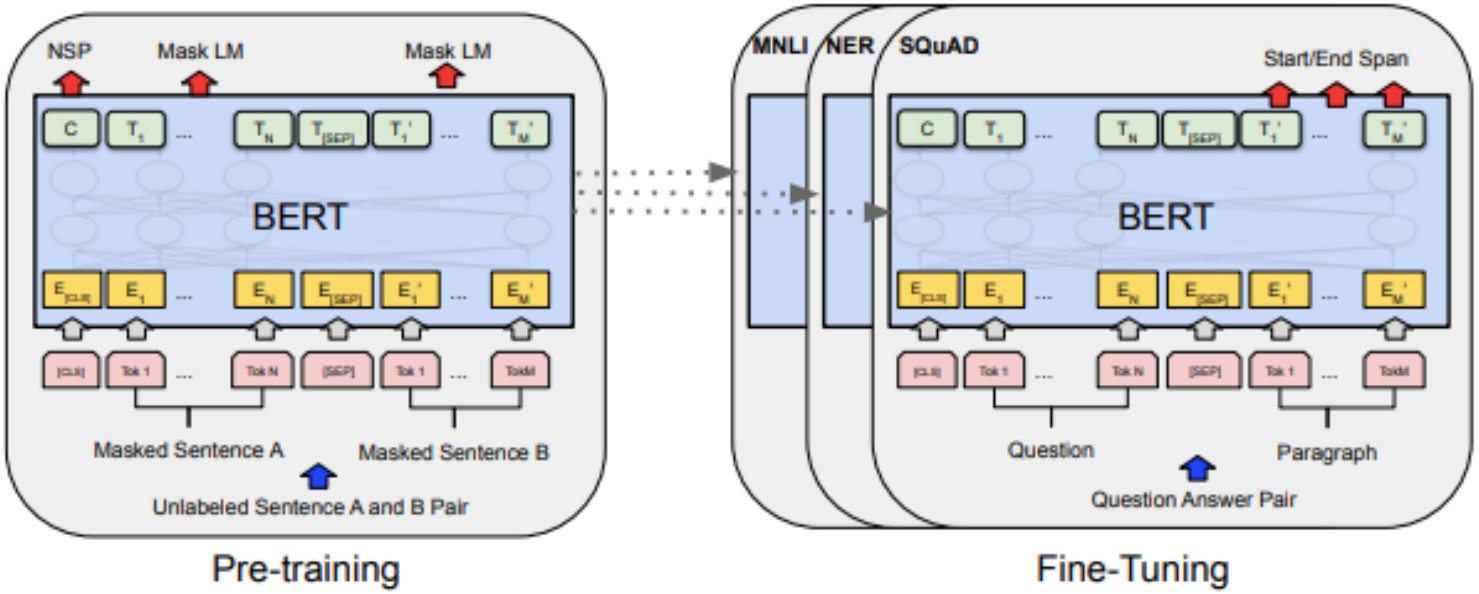
BERTは様々なタスクでSotAを更新したモデルである。

GLUEとは自然言語処理のためのタスクとデータセットの集合体です。
まとめると以下のようになっています。

通称	タスク名	タスク詳細
CoLA	The Corpus of Linguistic Acceptability	使われている英語が言語学的に受容されているか
SST-2	The Stanford Sentiment Treebank	映画の感想の感情分析
MRPC	Microsoft Research Paraphrase Corpus	オンラインニュースの文のペアが意味的に同じか
MNLI	MultiNLI Matched	含意関係に関するテキストデータ
QQP	Quora Question Pairs	2つの質問が同じかどうか
QNLI	Question NLI	質問と文は正しい答えを含んでいるかどうか
STS-B	Semantic Textual Similarity Benchmark	ニュース等の感想のペアが意味的に一致してるか
RTE	Recognizing Textual Entailment	含意関係に関するテキストデータ

BERTの概要

BERTは事前学習済みモデルであり、汎用的なモデルである。
画像認識でのVGG16やRESNET等のように、学習済みモデルからfine-tuning*1することで
少ないデータ量で目的に応じたタスクを学習することができ、とても汎用的です。



BERTの全体的な事前トレーニング(左)およびfine-tuning(右)。

*1 fine-tuningとは、事前学習したモデルの一部を再利用して、新しいモデルを構築する手法。

BERTの概要

BERTはWikipedia等の巨大なコーパスから教師なし学習で作成される。
自然言語処理ではアノテーションがかなり手間となり、大量のデータでの
教師あり学習は現実的でない。BERTの事前学習は教師なし学習であるため、
大量のテキストさえあれば事前学習を実行できます。

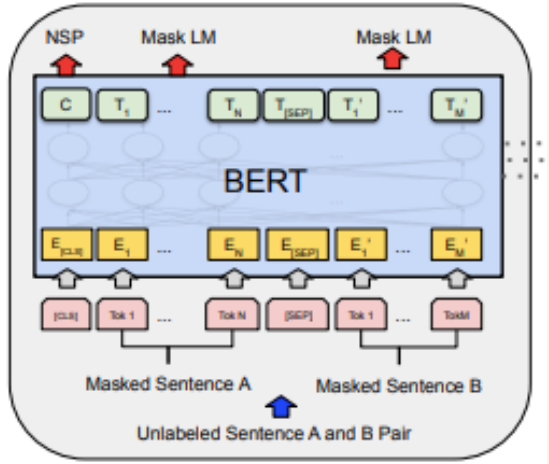
BERT (language model)

From Wikipedia, the free encyclopedia

Bidirectional Encoder Representations from Transformers (BERT) is a technique for NLP (Natural Language Processing) pre-training developed by Google. BERT was created and published in 2018 by Jacob Devlin and his colleagues from Google.[1][2] Google is leveraging BERT to better understand user searches.[3]

コーパス

教師なし学習
- マスクされた単語の予測(MLM)
- 次文予測(NSP)

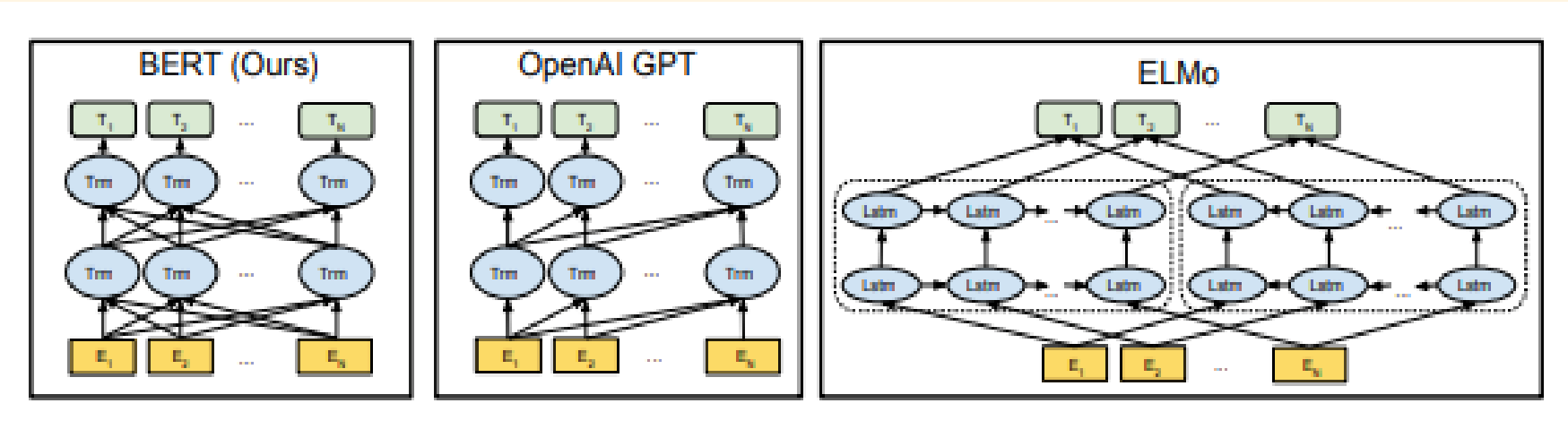


事前学習済みモデル

BERTの概要

BERTは文脈を双方向に学習するモデルである。

以前のモデルでは、前の単語から将来の単語を予測するLSTMやOpenAI GPT 、双方向ではあるが結びつきが弱いELMo等があったが、BERTでは単語を双方向から予測することが前提となっています。つまり、前後双方の単語を使用してタスクを解くことができます。



事前トレーニングモデルアーキテクチャの違い。 BERTは双方向トランスフォーマーを使用します。

BERTの概要

Nature Insight

BERTは周辺文脈を考慮して特徴量を決定できる。

BERTはAttentionという機構を持ち、周辺文脈を考慮して特徴量を決定します。
そのため、同じ単語でも文脈が異なれば、特徴量も異なります。

例えば、以下の文章を考えます。

- あの人は野球がうまい
- このラーメンはうまい

2つの文章の「うまい」は「上手い」と「美味い」で別の意味です。

BERTでは「うまい」という特徴量を文脈から作成するので、別の単語として認識して特徴量を作成することができます。

2. 事前学習タスク

事前学習タスク

事前学習タスク概要

BERTは事前学習済みモデルであり、事前学習で何を行うかが重要です。
BERTでは事前学習で手間をかけないよう、教師なし学習で事前学習を行います。
教師なし学習にすることで、大量のテキストデータさえあれば
人手でアノテーションを行うことなく学習をすることが可能となります。

BERTでは以下の2つのタスクを解きます。

- マスクされた単語の予測(MLM)
- 次文予測(NSP)

BERTの初出論文では、事前学習のトレーニングのコーパスは以下を使用しています。

- BooksCorpus（800Mワード）を使用（Zhu et al. 2015）
- 英語版ウィキペディア（2,500Mワード）

事前学習タスク

Nature Insight

Mask Language Model(MLM)

教師なし学習タスクの1つ目は、「マスクされた単語の予測(MLM)」です。
MLMはある単語をMASKさせて、周辺の単語からMASKした単語を予測する穴埋め問題です。
全体の15%の単語をマスクし、残りの単語の並びからマスクされた単語を予測します。
また、 pre-trainingで[MASK]トークン*は使用しないため、選択された全ての単語を
マスクするのではなく、マスクされた単語を以下のルールで置き換えます。

1. 80%はそのままマスクする。
例) my dog is hairy → my dog is [MASK]
2. 10%をランダムなトークンに置き換える。
例) my dog is apple
3. 10%をマスクせずに学習として使用する。
例) my dog is hairy

Masking Rates			Dev Set Results		
MASK	SAME	RND	MNLI	NER	
			Fine-tune	Fine-tune	Feature-based
80%	10%	10%	84.2	95.4	94.9
100%	0%	0%	84.3	94.9	94.0
80%	0%	20%	84.1	95.2	94.6
80%	20%	0%	84.4	95.2	94.7
0%	20%	80%	83.7	94.8	94.6
0%	0%	100%	83.6	94.9	94.6

図: MASKルールによる精度の比較

* トークン：BERTの入力に使用する単位。詳しくはP20で説明。

事前学習タスク

Nature Insight

Next Sentence Prediction(NSP)

教師なし学習タスクの2つ目は、「次文予測(NSP)」です。
質問応答(QA)や自然言語推論(NLI)などのタスクは、2つの文の関係を理解することが必要です。
文の関係性を理解するモデルをトレーニングするために、
単語コーパスから生成できる、2値化された次文予測タスクの事前学習を行います。

具体的には、以下のように学習データを作成し、次文予測を行います。

- 50%でBがAの後に続く実際の次文（としてラベル付け）
- 50%がコーパスからのランダムな文（としてラベル付け）

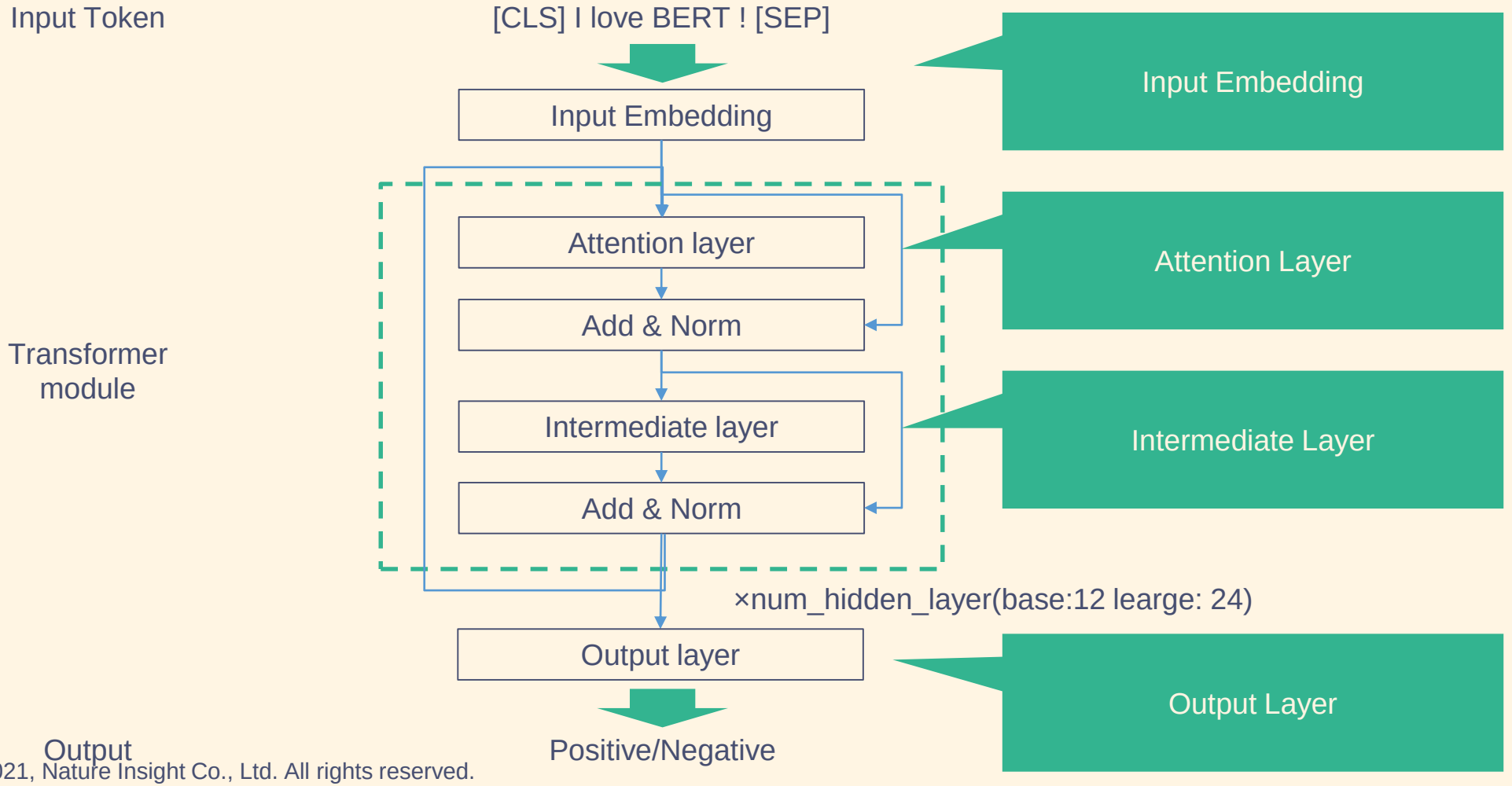
Input	label
[CLS] the man went to [MASK] store [SEP] he bought a gallon [MASK] milk [SEP]	IsNext
[CLS] the man [MASK] to the store [SEP] penguin [MASK] are flight ##less birds [SEP]	NotNext

3. BERTの中身

BERTの中身

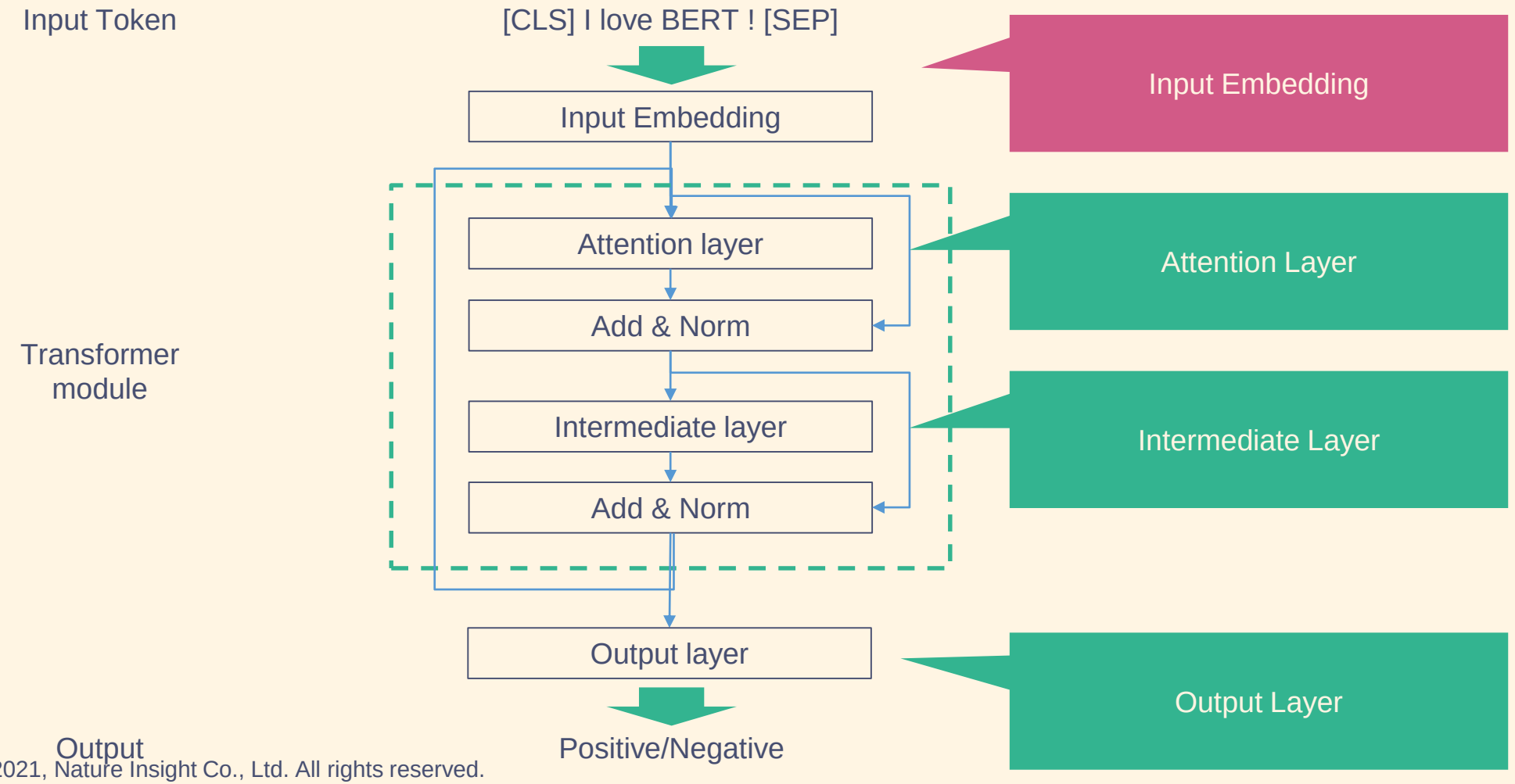
BERTの全体像

BERTでは、分かち書きされたトークンを入力とし、様々なタスクに応じたOutput layerを使用して出力します。この講義では次の順で説明します。



BERTの中身(Input Embedding)

この章では、BERTの肝であるInput Embeddingについて解説します。



BERTの中身(Input Embedding)

Nature Insight

BERTの入力：MeCab+Word Piece

入力の際に分かち書きをする際、通常の形態素解析器やスペース区切り等で作成した場合、語彙数が非常に大きくなります。

そこで、BERTではWordPieceというというサブワード化手法を使用して語彙数を減らします。これは、単語の中でさらに細かく分割し、高頻度の単語は1つの入力として扱い、 低頻度の単語はさらに細かく分割します。このようにして分割したものをトークンと呼びます。

日本語の場合は、単語で分かち書きするためにMeCabを合わせて使用します。

入力 [23]:
```
# tokenizerの分割例
text = "カーネーションが綺麗だった。"
tokenizer.tokenize(text)
```

出力[23]:
```
['カーネ', '##ーション', 'が', '綺麗', 'だっ', 'た', '。']
```

BERTの中身(Input Embedding)

Nature Insight

Input Embedding

BERTでは、Inputの先頭に[CLS]トークンを、文の終わりに[SEP]トークンを追加します。
また、Input Embeddingでは以下の3つのInput Embeddingを作成し、足し合わせます。

- Token Embedding
- Segment Embedding
- Position Embedding

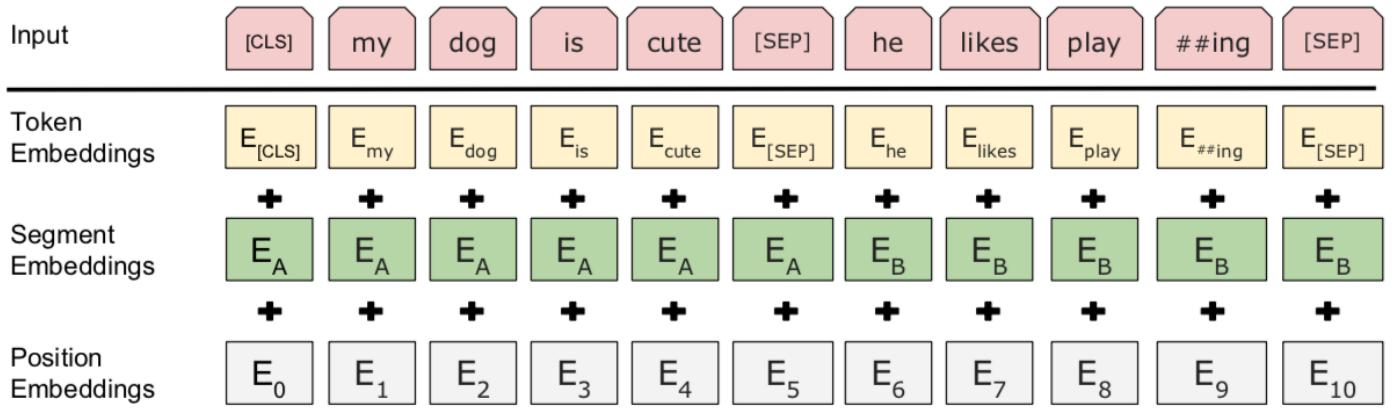


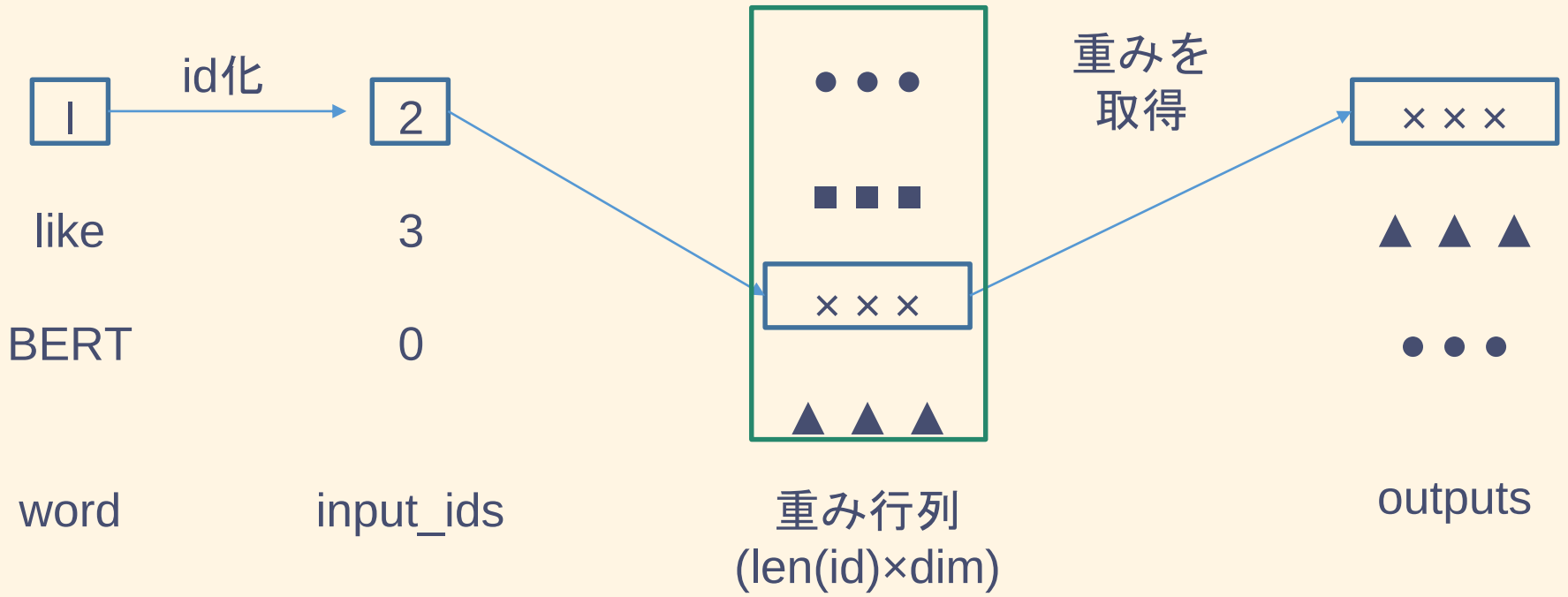
Figure 2: BERT input representation. The input embeddings is the sum of the token embeddings, the segmentation embeddings and the position embeddings.

図: 4-24 Input Representation(2文入れる場合)

BERTの中身(Input Embedding)

Token Embedding

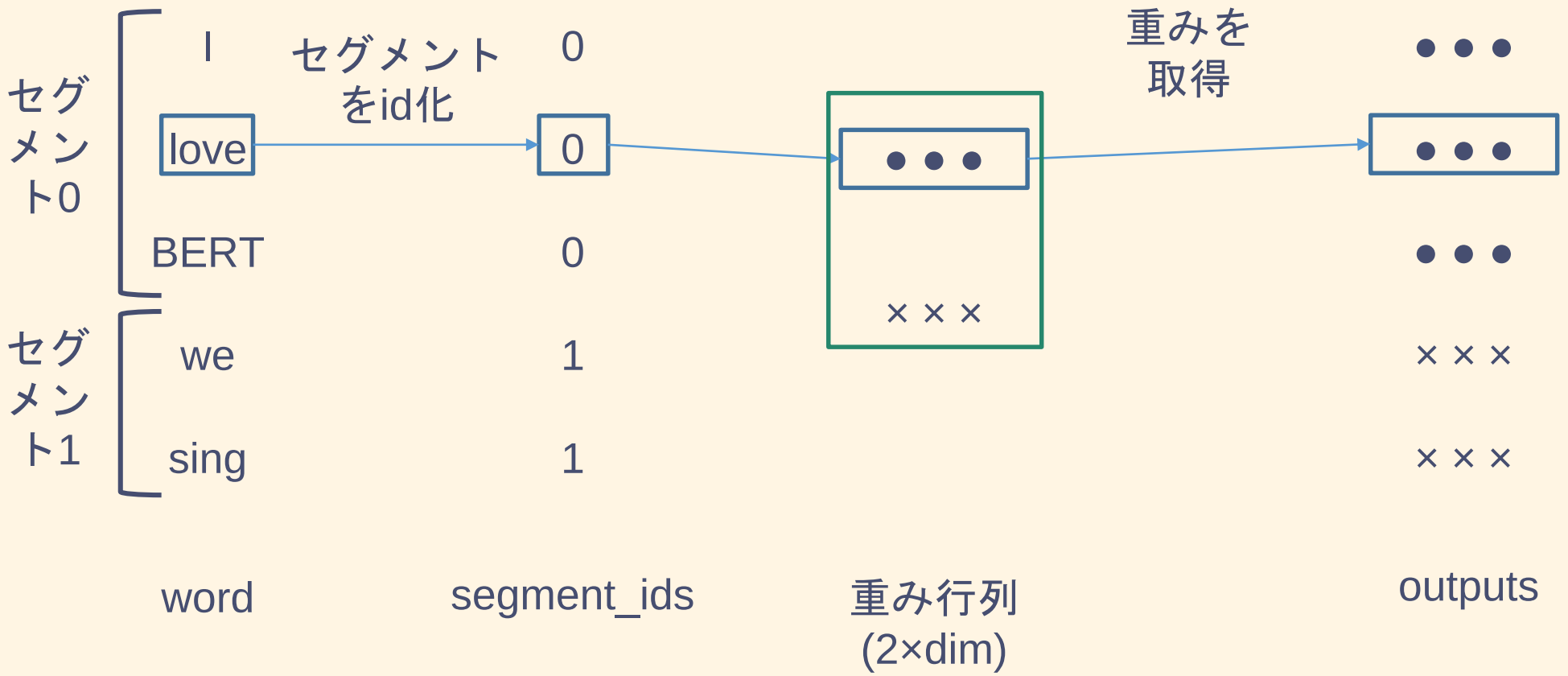
Token Embeddingでは、単語のidから特徴量を作成します。
実際の計算では、one-hot-encodingしてから重みを掛ける処理を行います。
(この層の仕組みを一般にEmbedding Layerと呼びます。)



BERTの中身(Input Embedding)

Segment Embedding

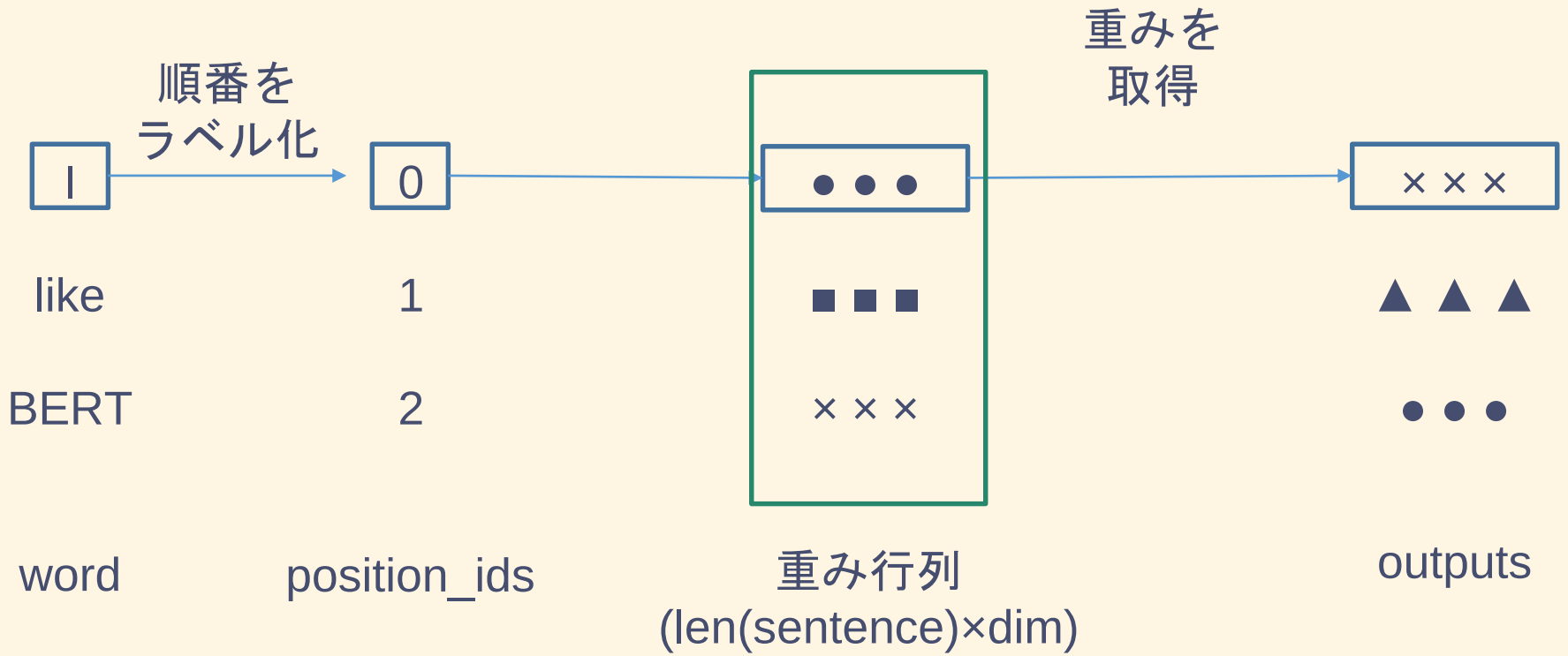
Segment Embeddingでは、文章のセグメントを0か1で入力します。
感情分析のように、文章を分けなくても良い場合は全て0を入力します。



BERTの中身(Input Embedding)

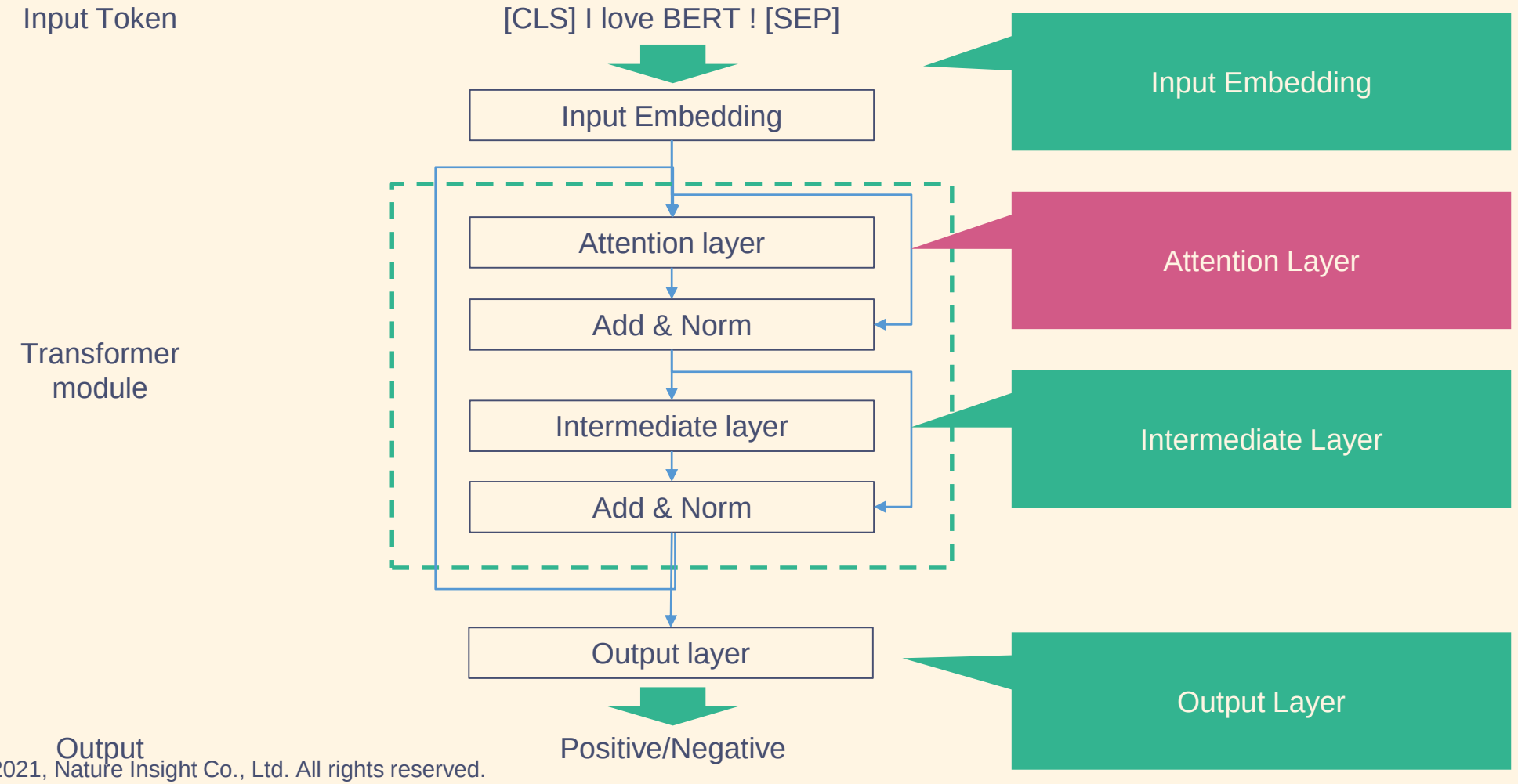
Positional Embedding

Positional Embeddingでは、単語の順番を0から順に入力します。
これにより単語がどの位置にあるかを識別できるようにします。



BERTの中身(Attention Layer)

この章では、BERTの肝であるAttention Layerについて解説します。



BERTの中身(Attention Layer)

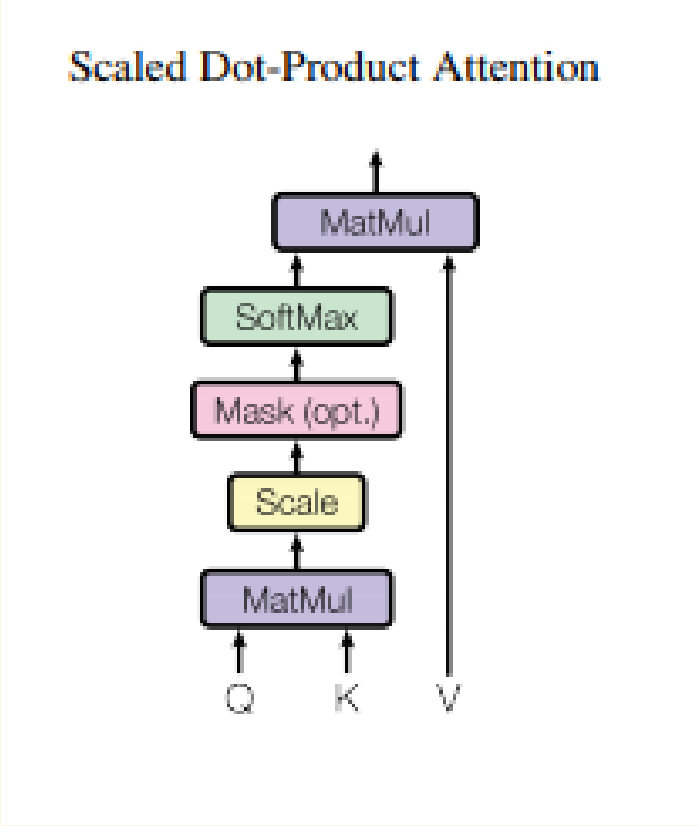
Scale Dot-Product Attention1

BERTで使用されているのはScaled Dot-Product Attentionです。
Q(Query), K(Key), V(Value)の3つの行列を使用して、トークン同士の影響度を考慮します。
Attention Layerを挟むことにより、同じ単語でも文章によって異なる特徴量を算出できます。
BERTでは、全て同一のInputから全結合($dense$)を通してQ, K, Vを生成します。
(全て同一のInputを使用するAttentionをSelf Attentionという。)

$$Q = dense_Q(Input)$$
$$K = dense_K(Input)$$
$$V = dense_V(Input)$$

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V$$

※Q, K, Vは全て(単語数×隠れ層の次元数)の行列



BERTの中身(Attention Layer)

Scale Dot-Product Attention2

さて、一つずつ解説していきましょう。まずは、MatMulとScaleの部分です。簡単に言えば、QとKの内積を計算してスケーリングすることで、QとKの類似度を算出しているようなイメージです。なお、dim1…dimdは、d次元の特徴量を表しています。(BERT baseではd=768)

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Q

	word	dim1	…	dimd
q_1	I	0.6	…	1
q_2	love	0.7	…	-0.1
q_3	BERT	0.2	…	0.3

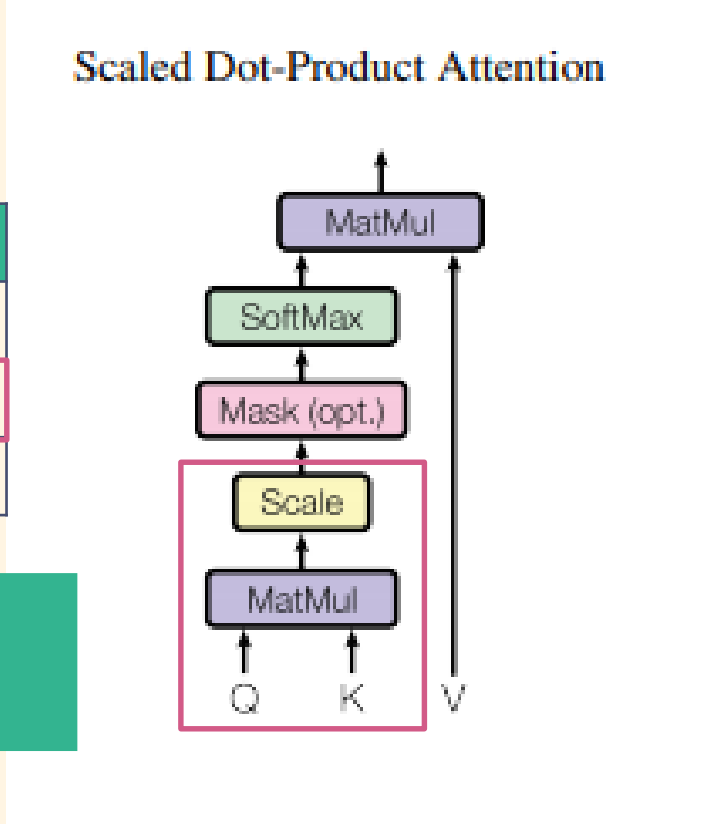
K

	word	dim1	…	dimd
k_1	I	0.5	…	0.7
k_2	love	0.6	…	-0.2
k_3	BERT	0.2	…	0.4

$\frac{QK^T}{\sqrt{d}}$

	I	love	BERT
I	1.2	-1.1	-1.0
love	-0.1	0.7	0.5
BERT	0.1	0.2	1.2

(2,3)成分は
q_2とk_3の内積を
スケーリングした値



BERTの中身(Attention Layer)

Scale Dot-Product Attention3

次に、$Softmax$の部分です。
これは横方向に$softmax$を取ることで、横を足し合わせて1になるように正規化しています。
この行列から、各トークンの、周辺のトークンによる重みを可視化することができます。

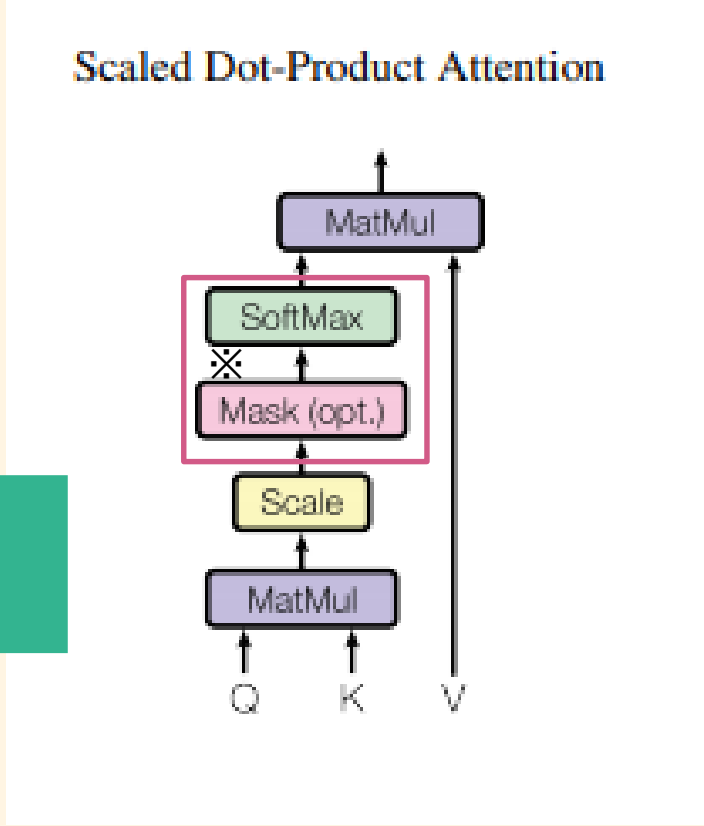
$$softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

	I	love	BERT
I	0.83	0.08	0.09
love	0.20	0.44	0.36
BERT	0.20	0.22	0.58

横に足して1

「love」は「BERT」に対して
36%の重みをつけると解釈



※MASKについて…BERTはPaddingを使用して可変長の文章の入力をしていますが、
Paddingした部分に-∞を入れることで、softmax値を0にしています。

BERTの中身(Attention Layer)

Scale Dot-Product Attention4

最後に、$Softmax$とVのMatMulです。
これは前述の$Softmax$を使って、Vの加重平均を出しています。
つまり、Attentionでは周辺のトークンの重みを考慮して、特徴量を生成していることになります。
これにより、同じトークンでも文章によって異なった特徴量を生成することが可能です。

$SoftMax$

	I	love	BERT
I	0.83	0.08	0.09
love	0.20	0.44	0.36
BERT	0.20	0.22	0.58

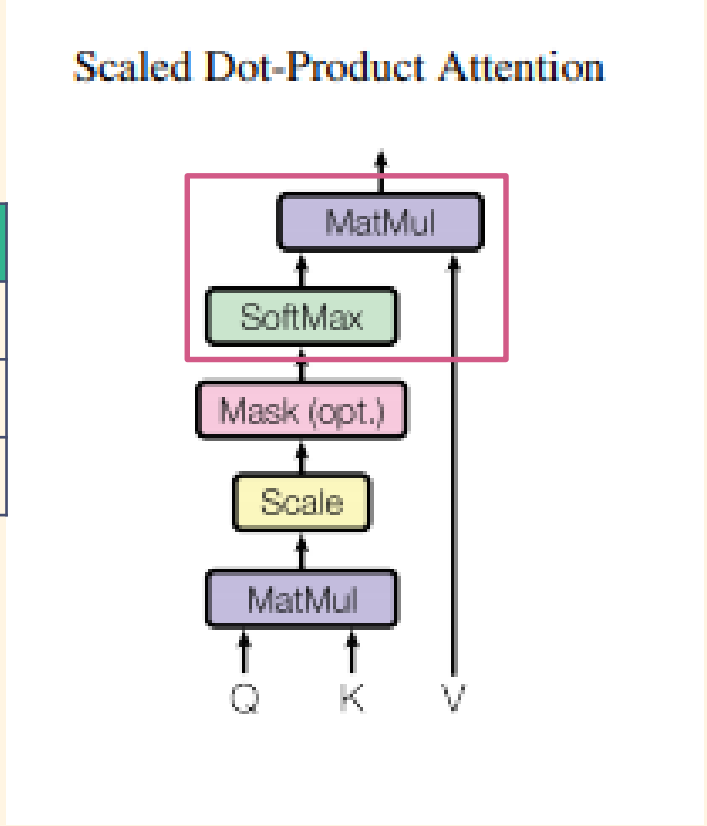
V

	word	dim1	…	dimd
v_1	I	0.4	…	0.8
v_2	love	0.2	…	-0.1
v_3	BERT	0.5	…	0.6

$SoftMax\,V$

	dim1	…	dimd
I	0.39	…	0.71
love	-0.35	…	0.33
BERT	0.41	…	0.49

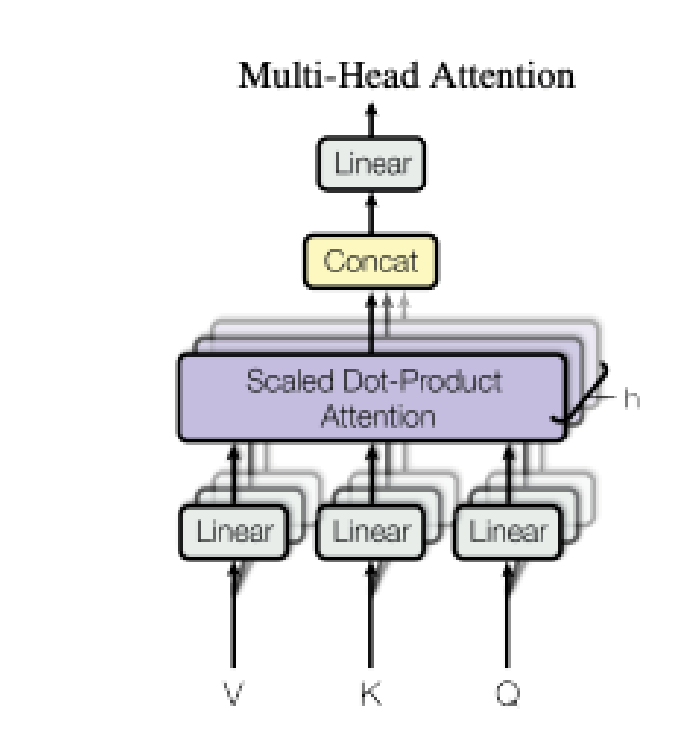
「BERT」のdim1は
SoftMaxの結果の
加重平均を取った値



BERTの中身(Attention Layer)

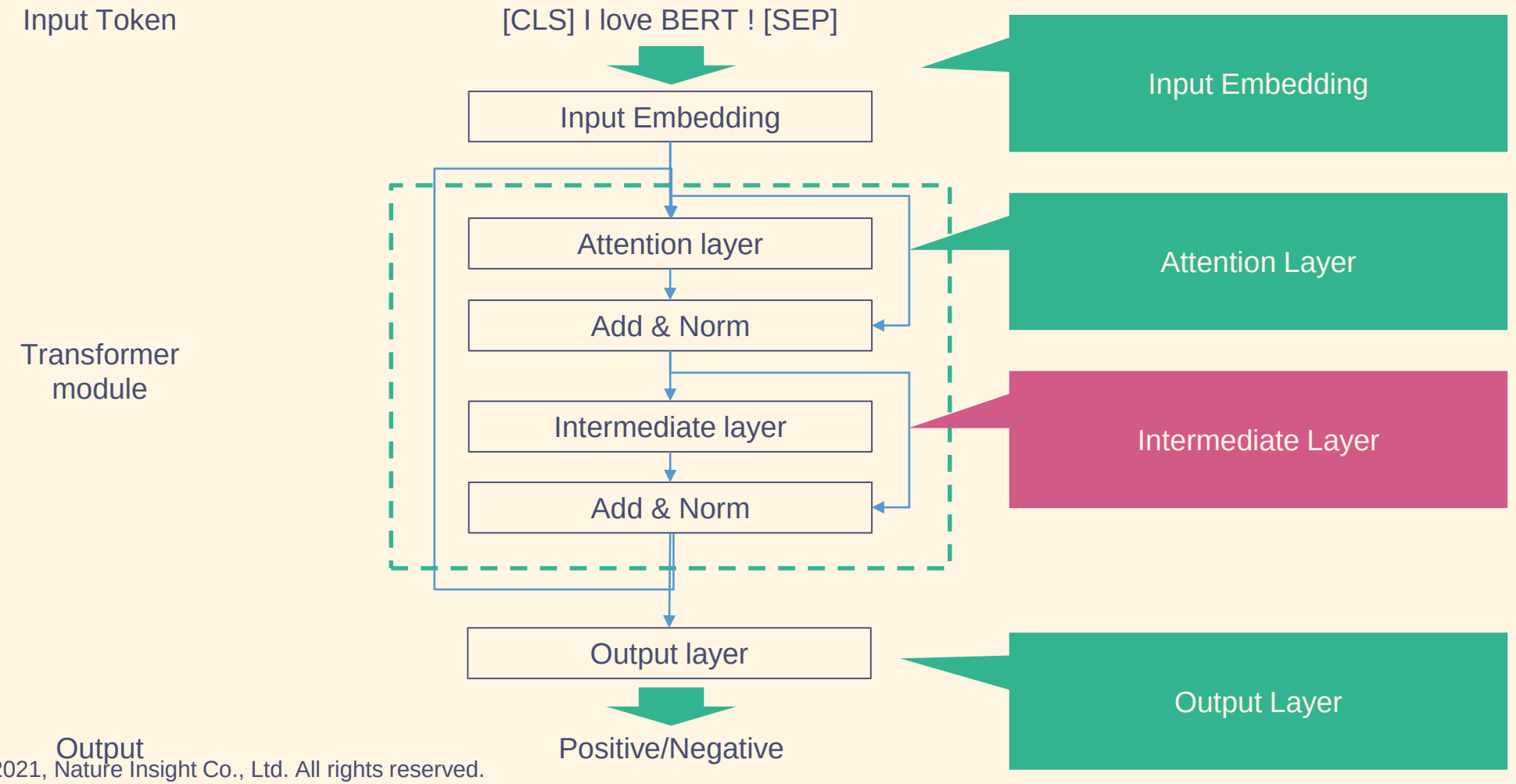
Malti-Head Attention

BERTでは、1つのAttentionではなく、特徴量の次元をh等分して複数のAttention Layerを生成しています。これをMulti-Head Attentionと呼びます。(BERTでは768次元を12等分しています。)
これにより、メモリの消費量も少なくなり、高速化も見込めます。



BERTの中身(Attention Layer)

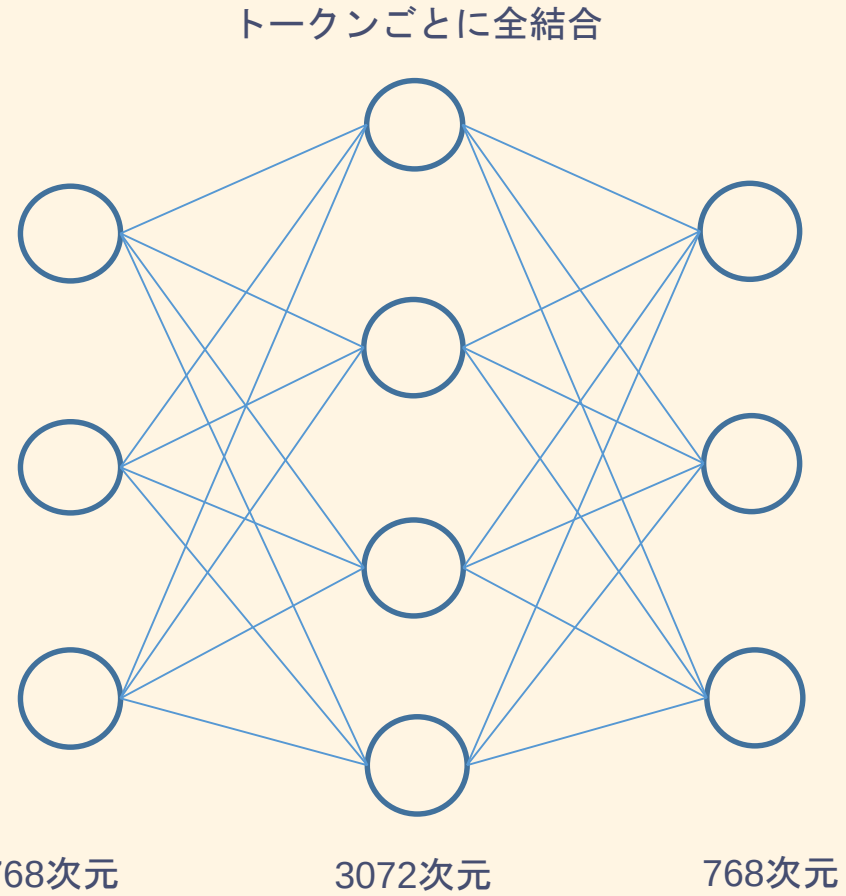
この章では、Attention layerの後につづくIntermediate layerを解説します。



BERTの中身(Intermediate Layer)

Intermediate Layer

Intermediate LayerはFeed Forword Networkとも呼ばれ、トークンごとに全結合を行う層です。
BERT baseでは、768次元を3072次元まで広げた後に、また768次元に戻しています。



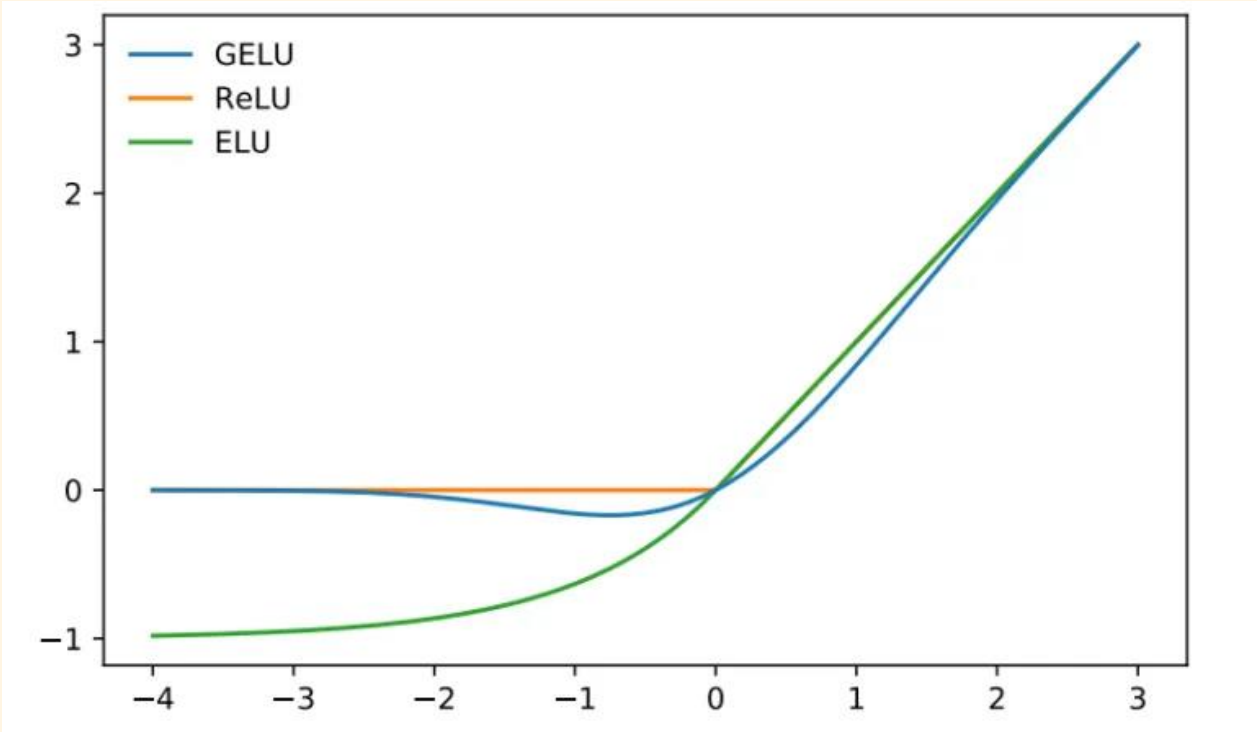
BERTの中身(Intermediate Layer)

Nature Insight

gelu関数

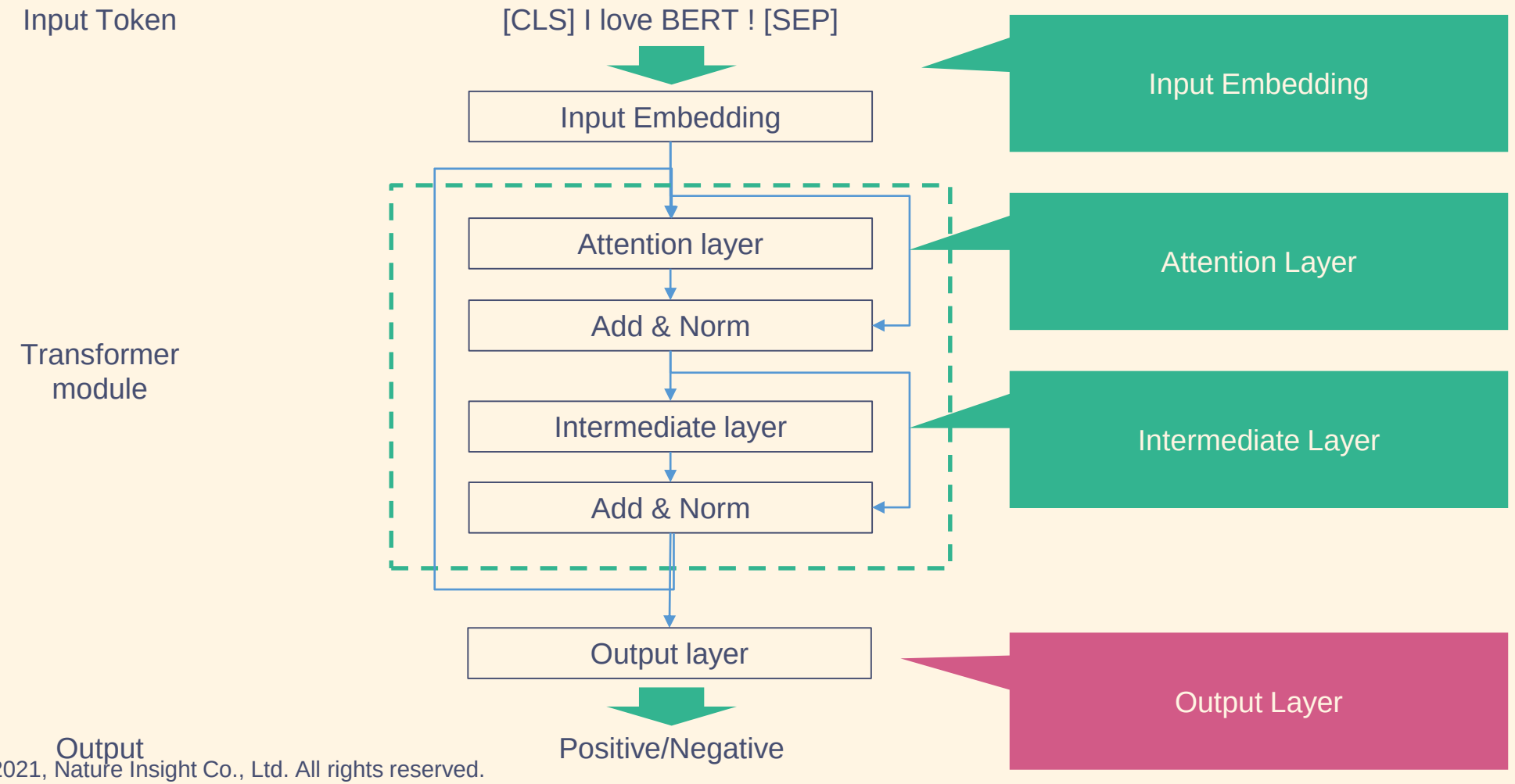
BERTのIntermediate Layerでは活性化関数にgeluを用いています。
geluはreluに近い関数ですが、0付近で滑らかになっていて、実数全体で微分可能です。

$$GELU(x) = xP(X \leq x)$$
where $X \sim N(0,1)$



BERTの中身(Output Layer)

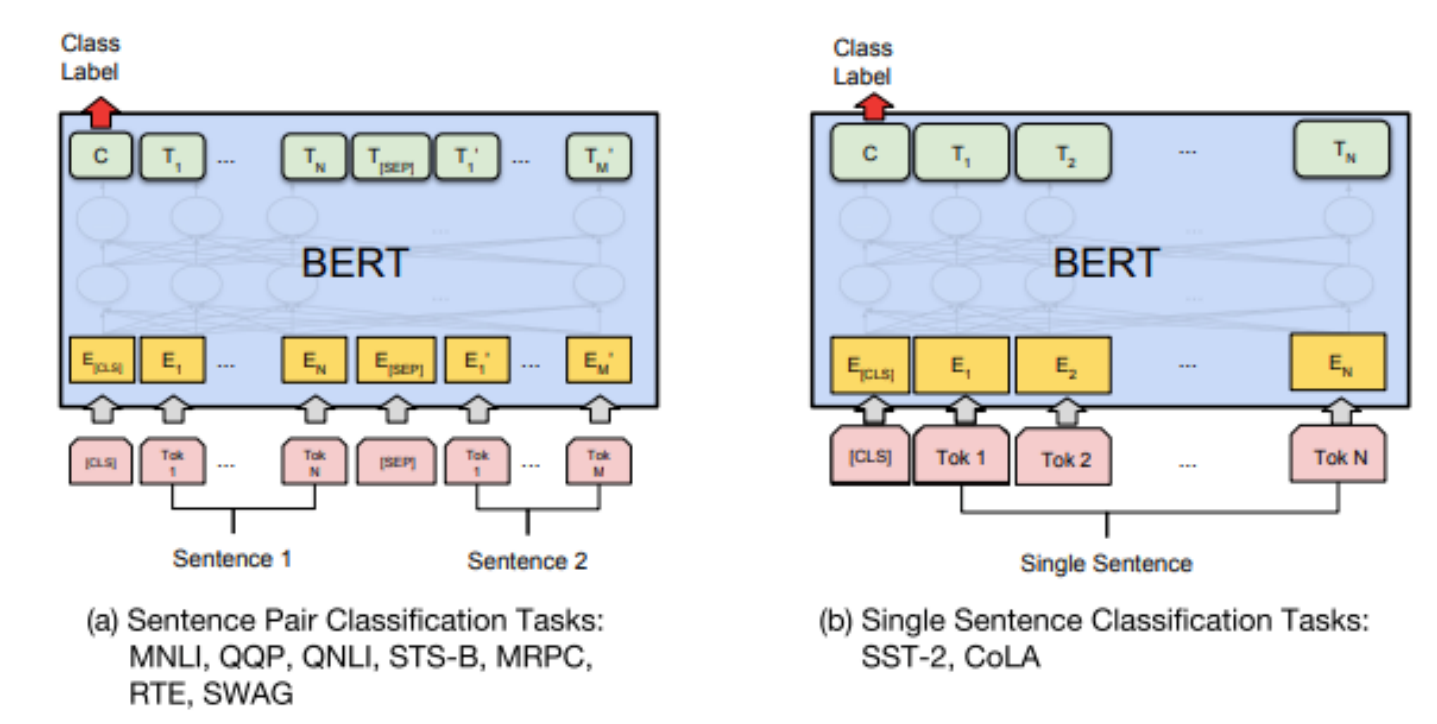
最後に、Output Layerを解説します。



BERTの中身(Output Layer)

文章のクラス分類

Output layerはタスクによって異なります。
次文予測(NSP)や感情分析等の文章ごとのクラス分類では、[CLS]トークンの特徴量を使用して全結合を行います。



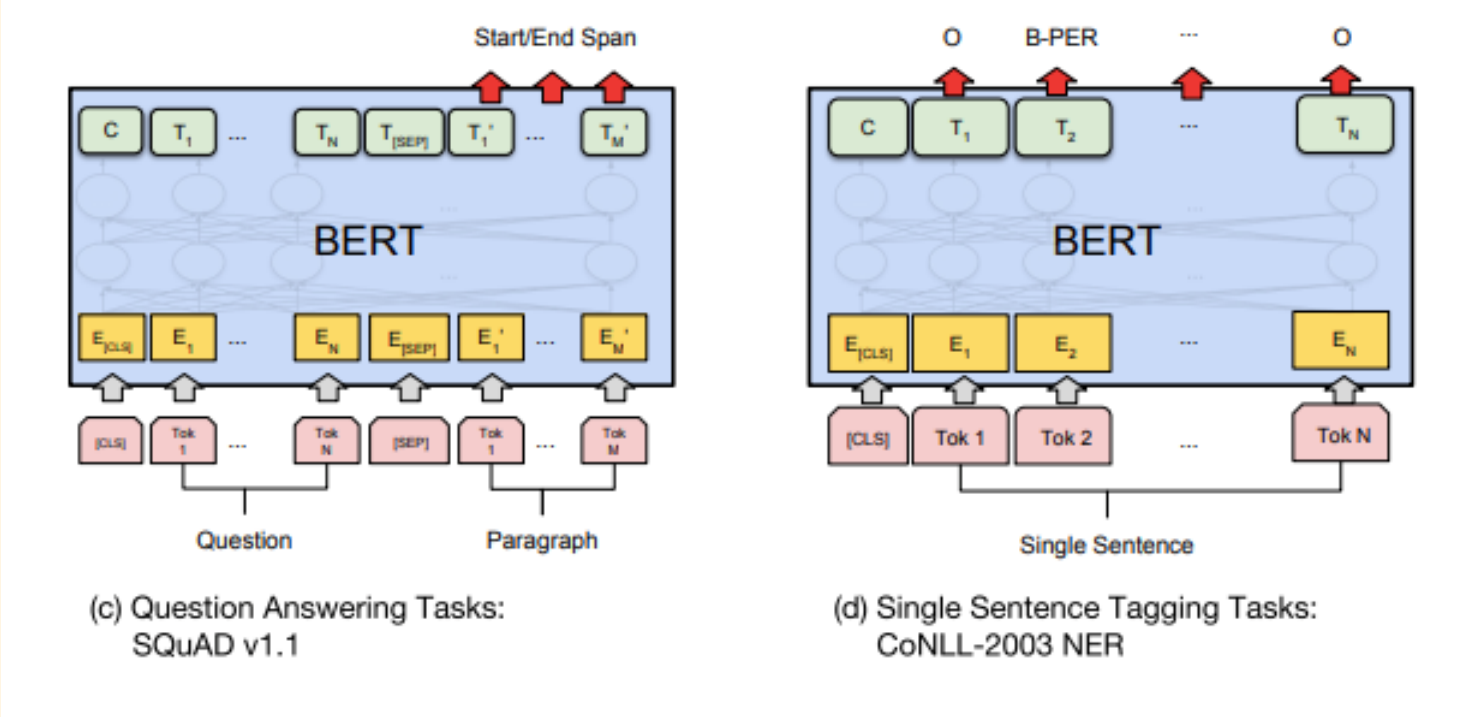
NSPなどのInputに
2セグメント使用するタスク

感情分析などのInputに
1セグメント使用するタスク

BERTの中身(Output Layer)

単語ごとのクラス分類

マスクされた単語予測(MLM)やQA、固有表現抽出などの単語ごとに出力する場合は、
トークンごと全結合を行い結果を出力します。



(c) Question Answering Tasks: SQuAD v1.1

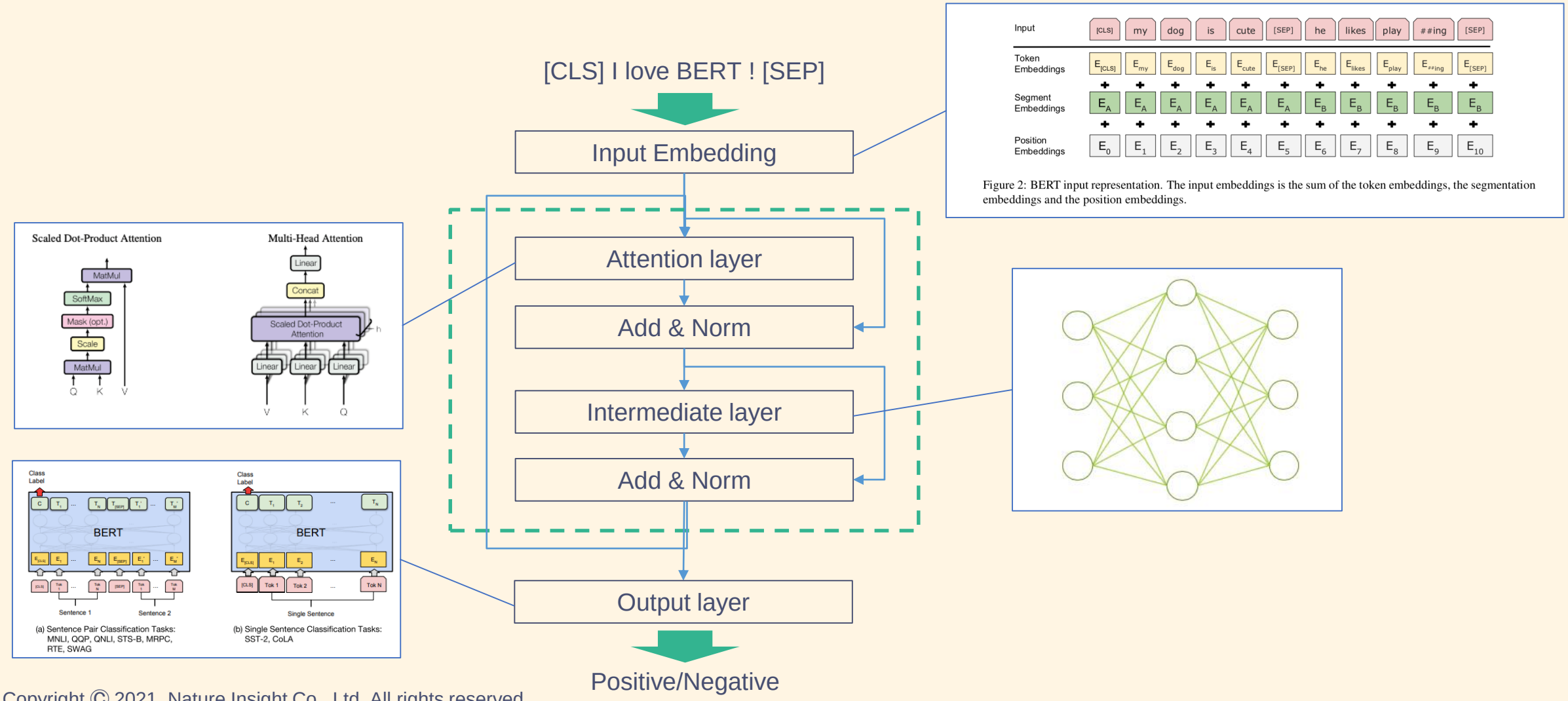
(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Q &AなどのInputに
2セグメント使用するタスク

MLMや固有表現抽出などのInputに
1セグメント使用するタスク

BERTの中身(まとめ)

まとめると以下のようになります。



4. BERTの使い方

BERTの使い方

BERTの使い方

BERTは事前学習済みモデルなので、Output Layerを変えることで様々なタスクを解くことができます。今回は、yahooレビューデータ(https://developer.yahoo.co.jp/sitemap/)を用いてポジネガ分析を行います。

コードは「3_BERT.ipynb」を参照してください。

yahooレビューデータは以下のようなデータです。Rating>3をポジティブ、Rating<3をネガティブとして扱います。(Rating=3は今回は使用しません。)

```
1 train_df[5:].head(5)
```

	Id	Place	CreateDate	Name	Subject	Body	Rating	Label
7	39dc2ef734d2e5c33aa3ef47a0b1288c1fa8c824	福岡	2009-02-25T12:00:32+09:00	インド料理ミナ	カレーの苦手な妻も絶賛	MINA大好きです♪かなりの評判のようですね～♪¥r¥r¥nお店の方も片言で親切です。思い出したら食べたくなりました♪今晩行こ～と♪¥r¥r¥nPS北九州にも最近MINAできたの？？	5.0	1
8	22ccf8229b2bed5a6602e9e0fc2e504e1c71d24d	大阪	2007-05-18T13:30:23+09:00	かに福	蟹とふぐが・・・	ふぐの専門店は多いけど、ふぐと蟹に特化しているお店って珍しい気がします。¥r¥r¥n私達はふぐの安いコースにプラス焼き蟹を注文しましたが、この組み合わせはとってもお薦めです。2Fの席に入りまし...	5.0	1
9	d8e79e2f507dd13bf44d2b44a5740e5fd010fb4f	宮城	2009-07-25T09:57:26+09:00	居酒屋 味人 AJITO	野球観戦の後に飲みに行くのに良いかも	Kスタ宮城から徒歩で7～8分の場所にあり、野球があった日は野球帰りのお客も多いが若い女性同士のお客も多く、落ち着いた雰囲気のお洒落な店です。料理はいずれも美味しかったし出てくるのも早かったです...	5.0	1
12	37a1f00dae7d688b9c0341fbf60cc815accb34b1	大阪	2014-02-23T06:05:33+09:00	株式会社ＡＢＣクラフト	接客は最悪	造花が置いてある入り口入ってすぐのアンティークっぽいガーデングッズを置いたコーナーで、気になったものがあり、在庫確認でそこのレジの方に声をかけたらものすごく嫌そうな、めんどくさそうに対応されま...	1.0	0
13	e7c9586d5e0b87acbc3b365742add666e4195736	北海道	2018-12-26T15:03:53+09:00	海へ 5条店	日本酒から大量のもずくの塊	日本酒から大量のもずくの塊がポソッと言う音とともにお猪口へ出て来た。店長に説明を求めると「言い訳になりますが、食洗器の中で伏せて調子を洗う際に、もずくの塊が入った」との事。¥r¥nそれって提供...	1.0	0

BERTの使い方

Nature Insight

BERTと事前学習済みモデル

概要でも述べましたが、BERTは、baseでもパラメータが約110Mあり、通常の学習では16TipのTPUで4日かかります。しかし、そのような膨大な計算資産、時間を用意するのは難しいです。

したがって、私たちが活用するには、事前学習済みモデルを使用して、計算コストを減らします。

事前学習済みモデルを使用すれば、少量の学習データと時間で、高い精度を出すことができます。

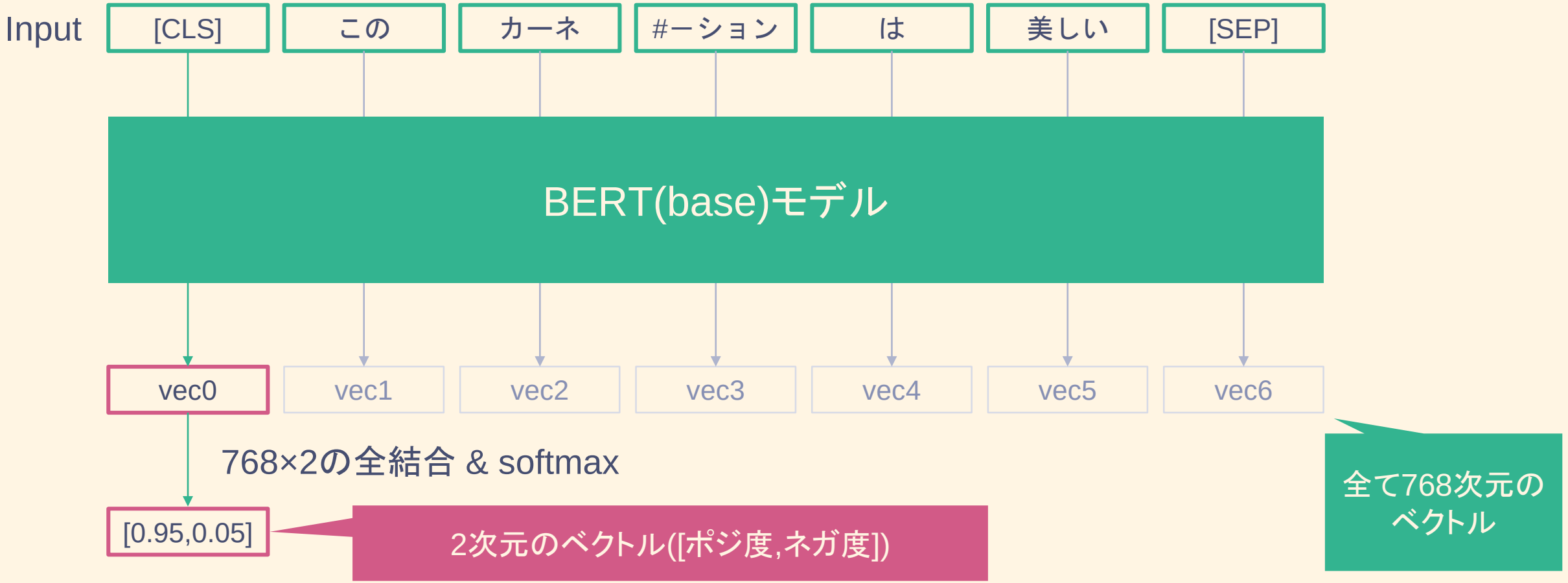
その際、事前学習済みモデルによって影響を受けるので、どんな学習データを使用しているのかは把握しておくことが必要です。

実際に実装する際は、以下のサイトのモジュールを使用することで、事前学習データを取得できます。(英語版、日本語版など、さまざまな言語があります。)
https://github.com/huggingface/transformers

BERTの使い方

BERTのポジネガモデル

BERT(base)のポジネガモデルでは、以下のようなモデルを使用します。



BERTの使い方

Nature Insight

BERTのポジネガモデル

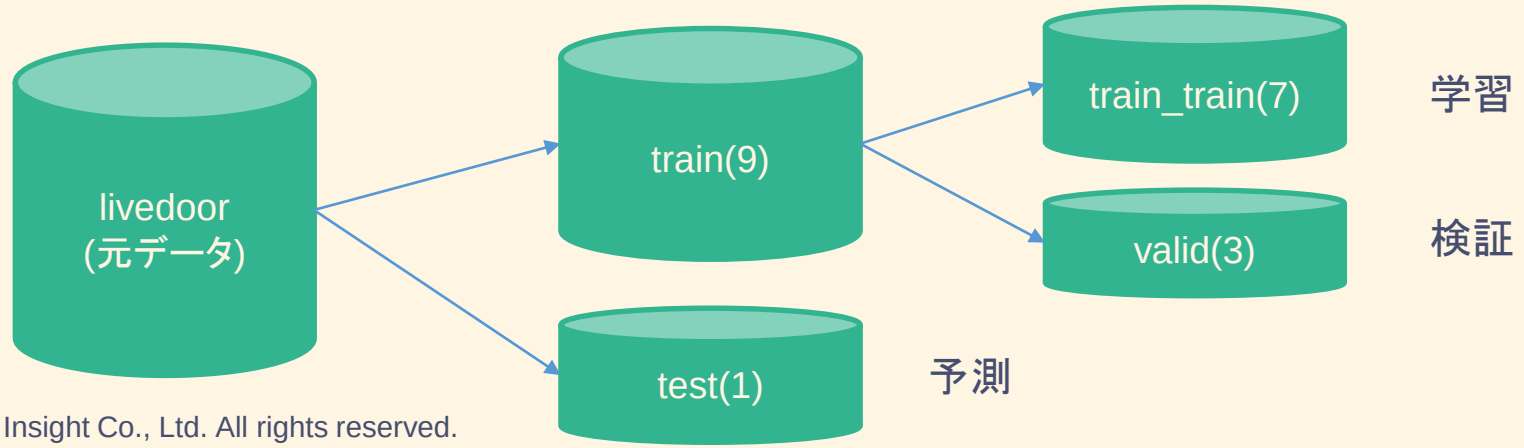
Yahooのレビューデータでポジネガを予測すると以下のようになります。

Text	Score
お店の雰囲気がとても良く、当日は同級生の飲み会で利用しましたが、デートで使えるな！！！と思いました。 赤鳥(？？)の串が絶品でした　**これ　おすすめです！！**	0.99030
姉の誕生日祝いで来ました。美味しい肉とエビが食べたいということで、以前来たとき、エビのグラタンが美味しかったことを思いだしこのお店に☆ **料理は美味しい**。ただ、**お店の接客態度が残念でした**。 ワインと一緒に♪と思っていたけど、ドリンクメニューは出してくれないし、料理の説明もなし。食べ残ってる食事を無言で下げられ…。ステーキソースが3種類出たけど、何のソースくらいは説明して欲しかったかなf(^_^; 期待してただけに(-_-)なによりお姉さんに申し訳ない。いくら料理が美味しくても。。。	0.23480
接客に問題ありという投稿がこれだけ多いのに、経営者や従業員は何も問題視してないんでしょうかね。 ランチ・セット頼んだんですが、デザートだけ来なくて・・・途中、早く持ってくるように頼んだのに、来ない。再度、ウェイトレスを呼ぶ・・・忘れていたらしい。 ウェイトレスといえども、日々の自分たちの仕事を問題意識を持って取り組んでいない証拠。客を人とは思っていない典型。結局、２時間ランチにかかってしまった。時間を返せ！！ やっつけ仕事、こなし仕事の店なのだろう。従業員教育やら社員教育なんて全然考えてもいない店なのだろう。 味や、コスパを論じる以前の問題。	0.07224

演習問題

演習問題

1. 「3.BERT.ipynb」で、BERTのInput、Outputなどを確認し、理解を深めましょう。
   また、パラメータ、イテレーション回数などを変更し、ポジネガの精度を高めましょう。
2. 「3.BERT.ipynb」 を参考に、livedoornews.csvのニュースの分類の予測を行うBERTモデルを、以下の条件を考慮して作成してみましょう。
   - 「body」列をテキスト、「media」列をターゲットとすること
   - mediaは9つのラベルがあるので、9値分類のBERTモデルを作成すること
   - 元のデータを9:1に分けて、 9の方をtrainに、1の方をtestにして、trainでBERTモデルを作成し、testに適用して予測すること
   - BERTモデルを作成する際は、trainの中でtrain_trainとvalidを7:3で分けて学習と検証を行うこと



参考文献

1. Attention Is All Your Need. (https://arxiv.org/abs/1706.03762)
2. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. (https://arxiv.org/abs/1810.04805)
3. つくりながら学ぶ! PyTorchによる発展ディープラーニング, 小川雄太郎 著

東京都千代田区神田小川町３-３
HF神田小川町ビルディング５階

03-3518-6061(代)

http://www.n-insight.co.jp/